

Weak Faddeev-Takhtajan-Volkov algebras; Lattice W_n algebras;

Farrokh Razavinia^{*1}

¹Department of discrete mathematics,
Moscow Institute of Physics and Technology (MIPT)

¹ f-razavinia@edu.hse.ru ;

f.razavinia@phystech.edu

February 27, 2017

†1 ‡ §

ABSTRACT. In this article, as always we will start by deliberating at our project's historical general view and then we will try to construct a new Poisson bracket on our simplest example sl_2 and then we will try to give a universal construction based on our universal variables and then will try to construct lattice W_2 algebras which will play a key role in our other constructions on lattice W_3 algebras and finally we will try to find the only non trivial dependent generator of our lattice W_4 algebras and so on for lattice W_n algebras.

And in the late of this article we will have appendix A, which will contain some parts of the Mathematica coding which we have used and have made for to find our algebra structures.

1. INTRODUCTION

This is an old project which has been considered and introduced by Boris Feigin in 1992. It has born in its new formulation on quantum Gelfand-Kirillov conjecture in a public conference at RIMS in 1992 based on the nilpotent part of $U_q(g)$ i.e. $U_q(\mathfrak{n})$ for g a simple Lie algebra.

Now, this problem is known as "Feigin's Conjecture".

In the mentioned talk, Feigin proposed the existence of a certain family of homomorphisms on quantized enveloping algebra $U_q(g)$ to the ring of skew-polynomials which will led us to a deffinition of lattice W -algebras. These "homomorphisms" has been turned to a very useful tool for to study the fraction field of quantized enveloping algebras. [6]

There been many attempt for to construct lattice W -algebras in Feigin's sence, which ensures the simplicity of the construction process of lattice W -algebra; for example the best known articles in the subject has been written by Kazuhiro Hikami and Rei Inoue who tried to obtain the algebra structure by using lax operators and generalized R matrices. [7] [8]

Or Alexander Belov and Alexander Antonov and Karen Chaltikian, who first tried to follow Feigin's construction but finaly they also solved part of the conjecture by getting help of lax operators, and it made very difficult to

^{*}Farrokh Razavinia

[†] The author would like to thank Professor Yaroslav Pugai for his helpful discussion during the preparation for this article.

[‡] subclass[2000]: Primary 16D10. 17B37. 81R50. ; Secondary 16W30. 20G42.

[§]keywords: Lattice W algebras, quantum groups, Feigin's homomorphisms, Boris Feigin, Yaroslav Pugai.

follow their publication.[9] [10]

But here in this article we will proceed and will introduce the most simplest way of constructing such kind of algebras by just employing Feigin's homomorphisms and screening operators by defining a Poisson bracket on our variables just based on our Cartan matrix. [1] [2]

In [2], Yaroslav Pugai has constructed lattice W_3 algebras already, but here we will introduce its weaker version based on a Poisson bracket as mentioned before, constructed on just Cartan matrix A_n , which will make our job more easier and more elegant.

For to do this, let us set C an arbitrary symmetrizable Cartan matrix of rank r and let $n = n_+$ be the standard maximal nilpotent sub-algebra of the Kac-Moody algebra associated with C .

So n is generated by elements E_1, \dots, E_r which are satisfying in Serre relations. [11] Where r stands by rank(C).

In [1], we proved that screening operators $S_{X_i^{ji}} = \sum_{\substack{j \in \mathbb{Z} \\ \text{for } i \text{ fixed}}}^n X_i^{ji}$; for X_i^{ji}

generators of the q -commutative ring $\mathbb{C}_q[X_i^{ji}] := \frac{\mathbb{C}[X_i^{ji}]}{X_i^{ji} X_k^{jk} - q^{<\alpha_i, \alpha_j>} X_k^{jk} X_i^{ji}}$ and for $<\alpha_i, \alpha_j> = a_{ij}$ the ij 's components of our Cartan matrix C ; are satisfying in quantum Serre relations $\text{ad}_q(X_i)^{1-a_{ij}}(X_j)$ for adjoint action $\text{ad}_q(X_i)(X_j) = X_i X_j - q^{a_{ij}} X_j X_i$ and $X_i \in (U_q)_\alpha$, $X_j \in (U_q)_\beta$. [5]

Where $(U_q)_\alpha = \{u \in U_q(g) | q^{\mathfrak{h}} u q^{-\mathfrak{h}} = q^{\alpha(\mathfrak{h})} u \text{ for all } \mathfrak{h} \in \check{P}\}$ and $U_q(g) = \bigoplus_{\alpha \in Q} (U_q)_\alpha$ for $Q = \bigoplus_{i \in I} \mathbb{Z} \alpha_i$ the root lattice and for \check{P} a free abelian group of rank $2|I| - \text{rank} A$ with \mathbb{Z} -basis $\{h_i | i \in I\} \cup \{d_s | s = 1, \dots, |I| - \text{rank} A\}$ and $\simeq = \mathbb{F} \otimes_{\mathbb{Z}} \check{P}$ be the \mathbb{F} -linear space spanned by \check{P} . [5] \check{P} will be called dual weight lattice and \simeq the Cartan subalgebra. And \mathbb{F} will stand for our ground field.[5]

Here for our Cartan matrix C , the quantum Serre relation will be

$$\begin{aligned} \text{ad}_q(X_i)^{1-(-1)}(X_j) &= \text{ad}_q^2(X_i)(X_j) \\ &= X_i^2 X_j - [2]_q X_i X_j X_i + X_j X_i^2 \\ &= X_i^2 X_j - (q + q^{-1}) X_i X_j X_i + X_j X_i^2 \end{aligned}$$

Where $[2]_q$ stands for quantum number $[n]_q = \frac{q^n - q^{-n}}{q - q^{-1}}$ in general.

And again as what we had in [1], we can define

$$U_q(n) := \langle S_{X_i^{ji}}, S_{X_k^{jk}} | (\text{ad}_q(S_{X_i^{ji}}))^2(S_{X_k^{jk}}) = 0 \rangle$$

and for $\mathbb{C}_q[X]$ the quantum polynomial ring in one variable and twisted tensor product $\bar{\otimes}$, we can define

$$\begin{aligned} U_q(n) \bar{\otimes} \mathbb{C}_q[X_l^{jl}] &:= \langle S_{X_i^{ji}}, S_{X_k^{jk}}, X_l^{jl} | (\text{ad}_q(S_{X_i^{ji}}))^2(S_{X_k^{jk}}) = 0 \\ &\quad, S_{X_i^{ji}} X_l^{jl} = q^2 X_l^{jl} S_{X_i^{ji}}, S_{X_k^{jk}} X_l^{jl} = q^{-1} X_l^{jl} S_{X_k^{jk}} \rangle \end{aligned}$$

such that we have the following embedding

$$U_q(n) \hookrightarrow U_q(n) \bar{\otimes} \mathbb{C}_q[X_l^{jl}] \hookrightarrow U_q(n) \bar{\otimes} \mathbb{C}_q[X_l^{jl}] \bar{\otimes} \mathbb{C}_q[X_m^{jm}]$$

where $\mathbb{C}_q[X_l^{jl}] \bar{\otimes} \mathbb{C}_q[X_m^{jm}] = \mathbb{C} \langle X_l^{jl}, X_m^{jm} | X_l^{jl} X_m^{jm} = q^{a_{lm}} X_m^{jm} X_l^{jl} \rangle$. [1]

Which will ensure the well definedness of our definition of lattice W -algebras.

2. WEAK FADDEEV-TAKHTAJAN-VOLKOV ALGEBRAS

As it has been mentioned already in [1], the main tools which we use are difference equations, screening operators, Feigin's homomorphisms and adjoint actions and partial differential equations and Cartan matrices and ...

We know that from an abstract view $g = sl_{m+1}$ is an algebra related to the Cartan matrix (a_{ij}) , where

$$a_{ij} = \begin{cases} 2 & \text{if } i = j \\ -1 & \text{if } |i - j| = 1 \\ 0 & \text{if } |i - j| > 1 \end{cases}$$

Now suppose that we have an infinite number of points in a definite discrete space such that we can assign them a proper coloring as follows
So by letting

$$A_n = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \vdots \\ 0 & -1 & 2 & -1 & 0 \\ \vdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}$$

be the Cartan matrix of sl_{m+1} for $n \in \{1, 2, \dots, m-1\}$, and so for sl_2 it will consist of just one row and one column, i.e. we have $A_1 = (2)$ and denote by $C[X]$ the skew polynomial ring on generators X_i labeled by $i \in \{-\infty, \dots, -1, 0, 1, \dots, +\infty\}$ and defining q -commutation relations

$$(2.1) \quad X_i X_j = q^2 X_j X_i \text{ for if } i \leq j$$

$$\overset{\circ}{X_{-\infty}} - \cdots - \cdots - \overset{\circ}{X_1} - \overset{\circ}{X_2} - \overset{\circ}{X_3} - \cdots - \cdots - \overset{\circ}{X_{+\infty}}$$

Definition 2.1. Let's define our Poisson bracket as follows in the case of sl_2 :

$$\begin{cases} \{X_i, X_j\} := 2X_i X_j & \text{if } i < j \\ \{X_i, X_i\} := 0 \end{cases}$$

The main problem is to find solutions of the system of difference equations from infinite number of non-commutative variables in quantum case and commutative variables in classical case. It is significant that commutation relations (2.1) depend on the sign of the difference $(i - j)$ only and is based on our Cartan matrix. We should try to find all solutions of the system:

$$(**) \begin{cases} \mathfrak{D}_x^{(n)} \triangleleft \tau_1 = 0 \\ H_x^{(n)} \triangleleft \tau_1 = 0 \end{cases}$$

And let us define our system of variables as follows

$$\begin{array}{cccccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & X_1^{(11)} & X_1^{(21)} & X_1^{(31)} & X_1^{(41)} & \cdots \\ \cdots & X_2^{(12)} & X_2^{(22)} & X_2^{(32)} & X_2^{(42)} & \cdots \\ \cdots & X_3^{(13)} & X_3^{(23)} & X_3^{(33)} & X_3^{(43)} & \cdots \\ \cdots & X_4^{(14)} & X_4^{(24)} & X_4^{(34)} & X_4^{(44)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

And let us equip this system of variables with lexicographic ordering, i.e. $j_{k_m} i < j_{k_n} i$ if $j_{k_m} < j_{k_n}$. And we need this kind of ordering because we have different kind of set of variables with a proper coloring such that each set has its own color different of its neighbors.

We have $\tau_1 := \tau_1[\cdots, X_1^{(11)}, X_1^{(21)}, X_1^{(31)}, \cdots, X_2^{(12)}, X_2^{(22)}, X_2^{(32)}, \cdots]$ is a multi-variable function depend on $\{x_i^{(ji)}\}$'s for $i, j \in \{-\infty, \cdots, 1, \cdots, n, \cdots, +\infty\}$ and $\mathfrak{D}_x^{(n)}$ comes from

$$(2.2) \quad \{S_{X_i^{ji}}, \tau_1\}_p = S_{X_i^{ji}} \tau_1 - p^{\deg \tau_1 \langle \alpha_i, \alpha_j \rangle} \tau_1 S_{X_i^{ji}}$$

where $\langle \alpha_i, \alpha_j \rangle = a_{ij}$ is related to our Cartan matrix and $S_{X_i^{ji}}$ is an screening operator on one of our variable sets, i.e. $S_{X_i^{ji}} = \sum_{j \in \mathbb{Z}} X_i^{ji}$. Then we will obtain the whole set of solutions by using the following shift operator:

$$\begin{aligned} \tau_2 &= \tau_1[X_1^{(11)} \rightarrow X_1^{(21)}, X_1^{(21)} \rightarrow X_1^{(31)}, \cdots], \\ (2.3) \quad \tau_3 &= \tau_2[X_1^{(21)} \rightarrow X_1^{(31)}, X_1^{(31)} \rightarrow X_1^{(41)}, \cdots] \\ &\vdots \end{aligned}$$

Definition 2.2. Let us define our lattice W-algebra based on its generators according to [2] [1]; Generators of lattice W-algebra associated with simple Lie algebra g constitute the functional basis of the space of invariants

$$(2.4) \quad \tau_i := \text{Inv}_{U_q(n_+)}(\mathbb{C}_q[X_i^{ji} | i \in \mathbb{Z}])$$

with additional requirements

$$(2.5) \quad H_{X_i^{ji}}(\tau_i) = 0 \quad \text{and} \quad D_{X_i^{ji}}(\tau_i) = 0$$

where $H_{X_i^{ji}}$ and $D_{X_i^{ji}}$ will be specified later.

Equation (2.4) means that the generators have to satisfy in quantum Serre relations and the first equation (2.5) means that they should have zero degree.

Here in this paper we just will work on $g = \mathfrak{sl}_n$ and will use $\tau_i^{(n)}$ instead of τ_i .

Where (n) sits for n in \mathfrak{sl}_n .

2.1. Lattice W_2 algebra. Let us first consider the sl_2 case for to open out the concepts of (2.2) and (2.4). And also for to simplifying out notations, let us consider our set of variables as $X_i := X_i^{ji}$.

And as it has shown in [1], it is enough just to work with $S_{X_i^{ji}} =: S_{X_i} = \sum_{i=1}^3 X_i$, because the other parts for $i > 3$ and $i < 1$ will tend to zero.

By setting $q = e^{-\hbar}$, for the Planck constant \hbar , we will try to find generators of our lattice W_2 -algebra, in the case of \mathfrak{sl}_2 .

First step:

First let us try to find $D_X^{(2)}$.

For to do this and for simplicity, we will set $\tau_1 := \tau_1[\cdots, X_1, X_2, X_3, \cdots]$.

And as it has been defined already, we have

$$\begin{aligned} D_X^{(2)} &:= \{S_{X_i}, \tau_1\} \\ &= \{X_1 + X_2 + X_3, \tau_1\} \\ &= \{X_1, \tau_1\} + \{X_2, \tau_1\} + \{X_3, \tau_1\} \\ (2.6) \quad &= (D_{X_1} + D_{X_2} + D_{X_3})\tau_1 \end{aligned}$$

Now for to understand what is (2.6), we note that partial $D_{X_i} = \{X_i, \tau_1\}$ and also note that our function $\tau_1[\cdots, X_1, X_2, X_3, \cdots]$ is a polynomial function consist of powers of X_i . What I mean is that, it is enough to find D_{X_i} on just powers of X_j for different values of $j \in \mathbb{Z}$.

So

$$(2.7) \quad (2.6) = \sum_j (\{X_1, X_j^n\} + \{X_2, X_j^n\} + \{X_3, X_j^n\})$$

Where according to rules which has been showed out in [1], we have

$$\begin{aligned} \{X_1, X_j^n\} &= X_1 X_j^n - q^{2n} X_j^n X_1 \\ &= \begin{cases} 0, & \text{if } j > 1 \\ (1 - q^{4n}) X_1 X_j^n, & \text{if } j < 1 \\ (1 - q^{2n}) X_1 X_j^n, & \text{if } j = 1 \end{cases} \end{aligned}$$

Where by setting $q = e^{-\hbar}$ and letting $\hbar = 1$ at the end, we will have:

First case: $j > 1$;

$$\{X_1, X_j^n\} = 0;$$

Second case: $j < 1$;

$$\begin{aligned}\{X_1, X_j^n\} &= (1 - e^{-4n\hbar})X_1, X_j^n \\ &\sim (1 - (1 - 4n\hbar))X_1, X_j^n \\ &= 4n\hbar X_1, X_j^n \sim 4nX_1, X_j^n \\ &= 4X_1 X_j \frac{\partial X_j^n}{\partial X_j}.\end{aligned}$$

Third case: $j = 1$;

$$\begin{aligned}\{X_1, X_1^n\} &= (1 - q^{2n})X_1 X_1^n \\ &= (1 - e^{-2n\hbar})X_1 X_1^n \\ &\sim (1 - (1 - 2n\hbar))X_1 X_1^n \\ &= 2n\hbar X_1 X_1^n \sim 2nX_1 X_1^n \\ &= 2X_1^2 \frac{\partial X_1^n}{\partial X_1}.\end{aligned}$$

And so we have

$$\begin{aligned}(2.7) &= \{X_1, X_1^n\} + \sum_{j<1} \{X_1, X_j^n\} + \sum_{j>1} \{X_1, X_j^n\} \\ &\quad + \{X_2, X_1^n\} + \sum_{j<2} \{X_2, X_j^n\} + \sum_{j>2} \{X_2, X_j^n\} \\ &\quad + \{X_3, X_1^n\} + \sum_{j<3} \{X_3, X_j^n\} + \sum_{j>3} \{X_3, X_j^n\} \\ &= 2X_1^2 \frac{\partial}{\partial X_1} + 0 + 0 \\ &\quad + 2X_2^2 \frac{\partial}{\partial X_2} + 4X_2 X_1 \frac{\partial}{\partial X_1} + 0 \\ &\quad + 2X_3^2 \frac{\partial}{\partial X_3} + 4X_3 X_2 \frac{\partial}{\partial X_2} + 4X_3 X_1 \frac{\partial}{\partial X_1} \\ &= 2X_1(X_1 + 2X_2 + 2X_3) \frac{\partial}{\partial X_1} + 2X_2(X_2 + 2X_3) \frac{\partial}{\partial X_2} \\ &\quad + 2X_3^2 \frac{\partial}{\partial X_3}.\end{aligned}$$

So we found $D_X^{(2)}$ which is as follows and we can omit 2, because finally we will make the action equal to zero and we can cancel 2 out from both sides. So we have

$$(2.8) \quad D_X^{(2)} = X_1(X_1 + 2X_2 + 2X_3) \frac{\partial}{\partial X_1} + X_2(X_2 + 2X_3) \frac{\partial}{\partial X_2} + X_3^2 \frac{\partial}{\partial X_3}$$

Second step:

Now we will try to find $H_X^{(2)}$.

For to find $H_X^{(2)}$, we note that it resembles degree of our polynomial function. So if for example $H_X^{(2)}$ acts on $X_1^n X_2^m X_3^l$, then we should get $(n+m+l)$. So let us define

$$(2.9) \quad H_X^{(2)} := \sum_i X_i \frac{\partial}{\partial X_i}$$

and then we have;

$$\begin{aligned}H_X^{(2)}(X_1^n X_2^m X_3^l) &= (\sum_i X_i \frac{\partial}{\partial X_i})(X_1^n X_2^m X_3^l) \\ &= \sum_i X_i \frac{\partial X_1^n X_2^m X_3^l}{\partial X_i} \\ &= X_1 \frac{\partial X_1^n X_2^m X_3^l}{\partial X_1} + X_2 \frac{\partial X_1^n X_2^m X_3^l}{\partial X_2} + X_3 \frac{\partial X_1^n X_2^m X_3^l}{\partial X_3} \\ &= nX_1^n X_2^m X_3^l + mX_1^n X_2^m X_3^l + lX_1^n X_2^m X_3^l \\ &= (n + m + l)X_1^n X_2^m X_3^l.\end{aligned}$$

Which gives us

$$H_X^{(2)}(X_1^n X_2^m X_3^l) = (n + m + l)X_1^n X_2^m X_3^l$$

and in the other side we have

$$\begin{aligned} (n+m+l)X_1^n X_2^m X_3^l &= nX_1^n X_1^{n-1} X_2^m X_3^l + mX_1^n X_2^n X_2^{m-1} X_3^l + lX_1^n X_2^m X_3^{l-1} \\ &= X_1 \frac{X_2^m X_3^l \partial X_1^n}{\partial X_1} + X_2 \frac{X_1^n X_3^l \partial X_2^m}{\partial X_2} + X_3 \frac{X_1^n X_2^m \partial X_3^l}{\partial X_3} \\ &= X_1 \frac{\partial}{\partial X_1} + X_2 \frac{\partial}{\partial X_2} + X_3 \frac{\partial}{\partial X_3} \end{aligned}$$

Which gives us

$$(n + m + l)X_1^n X_2^m X_3^l = \sum_i X_i \frac{\partial}{\partial X_i}$$

And it shows that (2.9) is well defined.

Now the only thing which remains is just to find the solutions of the following system of 2-linear homogeneous equations in one unknown τ_1 :

$$(2.10) \quad \begin{cases} (X_1(X_1 + 2X_2 + 2X_3) \frac{\partial}{\partial X_1} + X_2(X_2 + 2X_3) \frac{\partial}{\partial X_2} + X_3^2 \frac{\partial}{\partial X_3}) \tau_1[\dots, X_1, X_2, X_3, \\ \dots] = 0, \\ (X_1 \frac{\partial}{\partial X_1} + X_2 \frac{\partial}{\partial X_2} + X_3 \frac{\partial}{\partial X_3}) \tau_1[\dots, X_1, X_2, X_3, \dots] = 0; \end{cases}$$

Now the goal is to find such $\tau_1[\dots, X_1, X_2, X_3, \dots]$ which satisfies in our system of equations (2.10).

The second equation ensures that the solution has degree 0 and also the partial differentials will fix us a multi-variable function dependent on just X_1, X_2, X_3 .

The system of PDEs (2.10) can be solved using the procedure described in Chapter V, Sec IV of [3]. And for more details please check out appendix A. And after all it became clear that the system (2.10) has only one functional dependent nontrivial solution:

$$(2.11) \quad \tau_1^{(2)}[X_1, X_2, X_3] = \frac{(X_1 + X_2)(X_2 + X_3)}{X_2(X_1 + X_2 + X_3)} = \frac{(\sum_{1 \leq i_1 \leq 2} X_{i_1}^{(1)})(\sum_{1 \leq i_1 \leq 2} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{1 \leq i_1 \leq 3} X_{i_1}^{(1)})}$$

And again as before, (2) goes back to 2 in Sl_2 and 1 is a default index which will be used later it for to employ shifting operator.

According to the number of variables, we will have two shifts and then everything will be in a loop.

So here in sl_2 case we have three solutions for our system of linear equations (2.10) which belong to the fraction ring of polynomial functions.

$$(2.12) \quad \begin{cases} \tau_1^{(2)}[X_1, X_2, X_3] = \frac{(\sum_{1 \leq i_1 \leq 2} X_{i_1}^{(1)})(\sum_{1 \leq i_1 \leq 2} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{1 \leq i_1 \leq 3} X_{i_1}^{(1)})}; \\ \tau_2^{(2)}[X_2, X_3, X_4] = \frac{(\sum_{2 \leq i_1 \leq 3} X_{i_1}^{(1)})(\sum_{2 \leq i_1 \leq 3} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{2 \leq i_1 \leq 4} X_{i_1}^{(1)})}; \\ \tau_3^{(2)}[X_3, X_4, X_5] = \frac{(\sum_{3 \leq i_1 \leq 4} X_{i_1}^{(1)})(\sum_{3 \leq i_1 \leq 4} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{3 \leq i_1 \leq 5} X_{i_1}^{(1)})}; \end{cases}$$

And as it already has mentioned we go to define our non-commutative Poisson algebra according to definition of Poisson brackets given by Poisson himself [4] with this difference that here we work on q -commutative

ring $\frac{\mathbb{C}[X_i^{ji}]}{X_i^{ji} X_k^{jk} - q^{\langle \alpha_i, \alpha_k \rangle} X_k^{jk} X_i^{ji}}$, based on the generators which are the solutions

of PDEs system (2.10).

For to do this we will use the following bracket based on

$$\tau_i^{(n)}[\dots, X_1, X_2, X_3, \dots] \quad \text{and} \quad \tau_j^{(n)}[\dots, X_1, X_2, X_3, \dots]$$

So we have to define our Poisson brackets as what comes in follow:

$$(2.13) \quad F_j^{(n)} := \{\tau_i^{(n)}, \tau_j^{(n)}\} = \sum_i \frac{\partial \tau_i^{(n)}}{\partial X_i} \sum_j \frac{\partial \tau_j^{(n)}}{\partial X_j} \{X_i, X_j\}$$

Where $\{X_i, X_j\}$ is our previously defined Poisson bracket on our set of variables.

For instance in the case of sl_2 we have

$$\begin{aligned} \{\tau_1^{(2)}, \tau_2^{(2)}\} &= \left(\frac{\partial \tau_1^{(2)}}{\partial X_1} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} \{X_1, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3} \{X_1, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_4} \{X_1, X_4\} \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_2} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} \{X_2, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3} \{X_2, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_4} \{X_2, X_4\} \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_3} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} \{X_3, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3} \{X_3, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_4} \{X_3, X_4\} \right) \\ &= \left(\frac{\partial \tau_1^{(2)}}{\partial X_1} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} (2X_1X_2) + \frac{\partial \tau_2^{(2)}}{\partial X_3} (2X_1X_3) + \frac{\partial \tau_2^{(2)}}{\partial X_4} (2X_1X_4) \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_2} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} (0) + \frac{\partial \tau_2^{(2)}}{\partial X_3} (2X_2X_3) + \frac{\partial \tau_2^{(2)}}{\partial X_4} (2X_2X_4) \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_3} \right) \left(\frac{\partial \tau_2^{(2)}}{\partial X_2} (-2X_3X_2) + \frac{\partial \tau_2^{(2)}}{\partial X_3} (0) + \frac{\partial \tau_2^{(2)}}{\partial X_4} (2X_3X_4) \right) \\ &= 2 \frac{X_1X_2^2X_3^2X_4(X_1 + X_2 + X_3 + X_4)}{(X_1 + X_2)^2(X_2 + X_3)^3(X_3 + X_4)^2} \end{aligned}$$

So we have

$$(2.14) \quad F_2^{(2)} = \{\tau_1^{(2)}, \tau_2^{(2)}\} = \frac{2X_1X_2^2X_3^2X_4(X_1 + X_2 + X_3 + X_4)}{(X_1 + X_2)^2(X_2 + X_3)^3(X_3 + X_4)^2}$$

And it is enough to find our brackets on just first generator, because then we are able to find other brackets based on the other generators, so for $\tau_3^{(2)}$ we have in a same process as follows

$$\begin{aligned} F_3^{(2)} &= \{\tau_1^{(2)}, \tau_3^{(2)}\} \\ &= \left(\frac{\partial \tau_1^{(2)}}{\partial X_1} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} \{X_1, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4} \{X_1, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5} \{X_1, X_5\} \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_2} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} \{X_2, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4} \{X_2, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5} \{X_2, X_5\} \right) \\ &+ \left(\frac{\partial \tau_1^{(2)}}{\partial X_3} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} \{X_3, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4} \{X_3, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5} \{X_3, X_5\} \right) \\ &= \left(\frac{\partial \tau_1^{(2)}}{\partial X_1} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} (2X_1X_3) + \frac{\partial \tau_3^{(2)}}{\partial X_4} (2X_1X_4) + \frac{\partial \tau_3^{(2)}}{\partial X_5} (2X_1X_5) \right) \end{aligned}$$

$$\begin{aligned}
& + \left(\frac{\partial \tau_1^{(2)}}{\partial X_2} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} (2X_2 X_3) + \frac{\partial \tau_3^{(2)}}{\partial X_4} (2X_2 X_4) + \frac{\partial \tau_3^{(2)}}{\partial X_5} (2X_2 X_5) \right) \\
& + \left(\frac{\partial \tau_1^{(2)}}{\partial X_3} \right) \left(\frac{\partial \tau_3^{(2)}}{\partial X_3} (0) + \frac{\partial \tau_3^{(2)}}{\partial X_4} (2X_3 X_4) + \frac{\partial \tau_3^{(2)}}{\partial X_5} (2X_3 X_5) \right) \\
(2.15) \quad & = \frac{-2X_1 X_2 X_3^2 X_4 X_5}{(X_1 + X_2)(X_2 + X_3)^2(X_3 + X_4)^2(X_4 + X_5)}
\end{aligned}$$

We have to note that we almost are done with our Poisson algebra in sl_2 case, but for our further plan i.e. to find our Volterra system, the differential-difference chain of non-linear equations

$$(2.16) \quad \begin{cases} H = \sum_i [\ln(\tau_i)]; \\ \dot{\tau}_j = \{\tau_j, H\} = \tau_j \times \sum_i \Gamma_i; \end{cases}$$

Where Γ_i stands for $\frac{\tau_1, \tau_i}{\tau_1 \tau_i}$ [2]. Which means that we have to write down the brackets $\{\tau_1, \tau_i\}$ in terms of their decompositions to τ_j 's for $1 \leq j \leq i$. So we need to write it as decomposition of our generators and it will be done by using the Mathematica coding which we have produced in Appendix C.

And the result will be as follows

$$(2.17) \quad \begin{cases} F_2^{(2)} = \{\tau_1^{(2)}, \tau_2^{(2)}\} = 2(1 - \tau_1^{(2)})(1 - \tau_2^{(2)})(-1 + \tau_1^{(2)} + \tau_2^{(2)}); \\ F_3^{(2)} = \{\tau_1^{(2)}, \tau_3^{(2)}\} = -2(1 - \tau_1^{(2)})(1 - \tau_2^{(2)})(1 - \tau_3^{(2)}); \\ F_i^{(2)} = \{\tau_1^{(2)}, \tau_i^{(2)}\} = 0 \end{cases} \quad \text{for } |i - 1| \geq 3;$$

This result are weaker than Faddeev-Takhtajan-Volkov algebra which has mentioned in [2] and if we continue this for sl_3 , then we will have again a weak version of what that has mentioned in [2].

2.2. Lattice W_3 algebra. In this case we will use the following defined Poisson bracket based on Cartan matrix $A_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$.

But for to do this according to our previous ordering and list of variables, let us for simplicity set our variables as follows

Set $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$.

Definition 2.3. Let's define our Poisson bracket as follows in the case of sl_3 :

$$(2.18) \quad \begin{cases} \{X_i, X_j\} := 2X_i X_j & \text{if } i < j; \\ \{Y_i, Y_j\} := 2Y_i Y_j & \text{if } i < j; \\ \{X_i, X_i\} := 0; \\ \{Y_i, Y_i\} := 0; \\ \{X_i, Y_j\} := X_i Y_j & \text{if } i > j; \\ \{X_i, Y_j\} := -X_i Y_j & \text{if } i \leq j; \end{cases}$$

As it comes out, here our set of variables will be as follows:

$$\begin{array}{cccccccccccccccccccc} \circ & - & \dots & - & \circ & - & \circ & - & \circ & - & \circ & - & \circ & - & \circ & - & \dots & - & \circ & - & \circ & - & \circ & - & \circ & - & \dots & - & \circ \\ -\infty & & & & X_1 & Y_1 & X_2 & Y_2 & X_3 & Y_3 & & & & & & & & & X_{n-2} & Y_{n-2} & X_{n-1} & Y_{n-1} & X_n & Y_n & & & & & +\infty \end{array}$$

And instead of (2.1) we will have the following q -commutation relations

$$(2.19) \quad \begin{cases} X_i X_j = q^2 X_j X_i & \text{if } i \leq j; \\ Y_i Y_j = q^2 Y_j Y_i & \text{if } i \leq j; \\ X_i Y_j = q^{-1} Y_j X_i & \text{if } i \leq j; \end{cases}$$

And we will get the following equations in a same manner as in \mathfrak{sl}_2 :

First case: $i < j$;

$$\begin{aligned} \{X_i, Y_j^n\} &= X_i Y_j^n - q^{-n} Y_j^n X_i \\ &= X_i Y_j^n - q^0 X_i X_j^n \\ &= 0 \end{aligned}$$

Second case: $i \geq j$;

$$\begin{aligned} \{X_i, Y_j^n\} &= X_i Y_j^n - q^{-n} Y_j^n X_i \\ &= (1 - q^{-2n}) X_i Y_j^n \\ &= (1 - e^{2n\hbar}) X_i Y_j^n \\ &\sim (1 - (1 + 2n\hbar)) X_i Y_j^n \\ &= -2n\hbar X_i Y_j^n \\ &\sim -2n X_i Y_j^n \\ &= -2 X_i Y_j \frac{\partial Y_j^n}{\partial Y_j} \end{aligned}$$

$$(2.20) \quad \begin{cases} \{X_i, X_j^n\} = 0 & \text{if } i \leq j; \\ \{X_i, X_j^n\} = 4 X_i X_j \frac{\partial X_j^n}{\partial X_j} & \text{if } i > j; \\ \{X_i, Y_j^n\} = 0 & \text{if } i < j; \\ \{X_i, Y_j^n\} = -2 X_i Y_j \frac{\partial Y_j^n}{\partial Y_j} & \text{if } i \geq j; \\ \{Y_j, X_i^n\} = -2 Y_j X_i \frac{\partial X_i^n}{\partial X_i} & \text{if } i \leq j; \end{cases}$$

According to (2.20) we will try to find $H_X^{(3)}$ as what comes in follow

$$\begin{aligned} &\{X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3}, X_0\} \\ &= X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} X_0 - X_0 X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} \\ &= (1 - q^{2\alpha_1+2\alpha_2+2\alpha_3-\beta_1-\beta_2-\beta_3}) X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} X_0 \\ &\sim (1 - (1 - n\hbar(2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3))) X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} X_0 \\ &= (2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3) n\hbar X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} X_0 \\ &\sim (2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3) n X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} Y_1^{\beta_1} Y_2^{\beta_2} Y_3^{\beta_3} X_0 \\ &= (2X_1 \frac{\partial}{\partial X_1} + 2X_2 \frac{\partial}{\partial X_2} + 2X_3 \frac{\partial}{\partial X_3} - Y_1 \frac{\partial}{\partial Y_1} - Y_2 \frac{\partial}{\partial Y_2} - Y_3 \frac{\partial}{\partial Y_3}) \tau_1^{(3)}. \end{aligned}$$

Now let us as usual suppose $i > j$ and then we will define the following same quantities

Here we have for X_i s:

$$\begin{aligned}
X_j D_{X_i} &:= \{X_i, X_j^n\} \\
&= X_i X_j^n - q^{2n} X_j^n X_i \\
&= (1 - q^{4n}) X_i X_j^n \\
&= (1 - e^{-4n\hbar}) X_i X_j^n \\
&\sim (1 - (1 - 4n\hbar)) X_i X_j^n \\
&= 4n\hbar X_i X_j^n \\
&\sim 4n X_i X_j^n \\
&= 4n X_i X_j \frac{\partial X_j^n}{\partial X_j}.
\end{aligned}$$

And the same will be for Y_i s.

And for the different quantities X_i and Y_j s we have:

First case: for $i > j$;

$$\begin{aligned}
Y_j D_{X_i} &:= \{X_i, Y_j^n\} \\
&= ad_{X_i} Y_j^n \\
&= X_i Y_j^n - q^{-n} Y_j^n X_i \\
&= (1 - q^{-2n}) X_i Y_j^n \\
&= (1 - e^{-2n\hbar}) X_i Y_j^n \\
&\sim (1 - (1 - 2n\hbar)) X_i Y_j^n \\
&= 2n\hbar X_i Y_j^n \\
&\sim 2n X_i Y_j^n \\
&= 2X_i Y_j \frac{\partial Y_j^n}{\partial Y_j}.
\end{aligned}$$

Second case: for $i \leq j$;

According to what has just mentioned we have

$$\begin{aligned}
Y_j D_1^Y &:=_{Y_j} D_{Y_1} \\
&= 4Y_1 Y_j \frac{\partial Y_j^n}{\partial Y_j}.
\end{aligned}$$

And

$$\begin{aligned}
Y_1 D_1^Y &:=_{Y_1} D_{Y_1} \\
&= 2Y_1^2 \frac{\partial Y_1^n}{\partial Y_1}.
\end{aligned}$$

And in a same way we can find the desired results for $Y_j D_2^Y$ and $Y_j D_3^Y$,
So let us define

$$(2.21) \quad \begin{cases} Y D_1^Y :=_{Y_1} D_1^Y +_{Y_j}^{j<1} D_1^Y +_{Y_j}^{j>1} D_1^Y; \\ Y D_2^Y :=_{Y_2} D_2^Y +_{Y_j}^{j<2} D_2^Y +_{Y_j}^{j>2} D_2^Y; \\ Y D_3^Y :=_{Y_3} D_3^Y +_{Y_j}^{j<3} D_3^Y +_{Y_j}^{j>3} D_3^Y; \end{cases}$$

And then we will have

$${}_Y D_1^Y = Y_1^2 \frac{\partial}{\partial Y_1} + \sum_{j<1} 2Y_1 Y_j \frac{\partial}{\partial Y_j} + 0$$

And

$${}_Y D_2^Y = Y_2^2 \frac{\partial}{\partial Y_2} + \sum_{j<2} 2Y_2 Y_j \frac{\partial}{\partial Y_j} + 0$$

And

$${}_Y D_3^Y = Y_3^2 \frac{\partial}{\partial Y_3} + \sum_{j < 2} 2Y_3 Y_j \frac{\partial}{\partial Y_j} + 0$$

And finally we get

$$\begin{aligned} {}_Y D_Y^{(3)} &:= {}_Y D_1 + {}_Y D_2 + {}_Y D_3 \\ &= Y_1(Y_1 + 2Y_2 + 2Y_3) \frac{\partial}{\partial Y_1} + Y_2(Y_2 + 2Y_3) \frac{\partial}{\partial Y_2} + Y_3^2 \frac{\partial}{\partial Y_3}. \end{aligned}$$

For $j \geq 1$ we have

$$\begin{aligned} {}_{X_1} D_{Y_j} &:= \{Y_j, X_1^n\} \\ &= Y_j X_1^n - q^{-n} X_1^n Y_j \\ &= (1 - q^{-2n}) Y_j X_1^n \\ &= (1 - e^{2n\hbar}) Y_j X_1^n \\ &\sim (1 - (1 + 2n\hbar)) Y_j X_1^n \\ &= -2n\hbar Y_j X_1^n \\ &\sim -2n Y_j X_1^n \\ &= -2Y_j X_1 \frac{\partial}{\partial X_1} \\ &= -2X_1(Y_1 + Y_2 + Y_3) \frac{\partial}{\partial X_1}. \end{aligned}$$

For $j \geq 2$ we have

$$\begin{aligned} {}_{X_2} D_{Y_j} &:= \{Y_j, X_2^n\} \\ &= Y_j X_2^n - q^{-n} X_2^n Y_j \\ &= (1 - q^{-2n}) Y_j X_2^n \\ &= (1 - e^{2n\hbar}) Y_j X_2^n \\ &\sim (1 - (1 + 2n\hbar)) Y_j X_2^n \\ &= -2n\hbar Y_j X_2^n \\ &\sim -2n Y_j X_2^n \\ &= -2Y_j X_2 \frac{\partial}{\partial X_2} \\ &= -2X_2(Y_2 + Y_3) \frac{\partial}{\partial X_2}. \end{aligned}$$

For $j \geq 3$ we have

$$\begin{aligned} {}_{X_3} D_{Y_j} &:= \{Y_j, X_3^n\} \\ &= Y_j X_3^n - q^{-n} X_3^n Y_j \\ &= (1 - q^{-2n}) Y_j X_3^n \\ &= (1 - e^{2n\hbar}) Y_j X_3^n \\ &\sim (1 - (1 + 2n\hbar)) Y_j X_3^n \\ &= -2n\hbar Y_j X_3^n \\ &\sim -2n Y_j X_3^n \\ &= -2Y_j X_3 \frac{\partial}{\partial X_3} \\ &= -2X_3 Y_3 \frac{\partial}{\partial X_3}. \end{aligned}$$

And after all these, let us define

$$\begin{aligned} {}_X D_Y^{(3)} &:= {}_{X_1} D_{Y_j} + {}_{X_2} D_{Y_j} + {}_{X_3} D_{Y_j} \\ &= -2X_1(Y_1 + Y_2 + Y_3) \frac{\partial}{\partial X_1} - 2X_2(Y_2 + Y_3) \frac{\partial}{\partial X_2} \\ &\quad - 2X_3 Y_3 \frac{\partial}{\partial X_3}. \end{aligned}$$

And finally let us define

$$\begin{aligned} D_Y^{(3)} &:= {}_Y D_Y^{(3)} + {}_X D_Y^{(3)} \\ &= Y_1(Y_1 + 2Y_2 + 2Y_3) \frac{\partial}{\partial Y_1} + Y_2(Y_2 + 2Y_3) \frac{\partial}{\partial Y_2} \\ &\quad + Y_3^2 \frac{\partial}{\partial Y_3} - 2X_1(Y_1 + Y_2 + Y_3) \frac{\partial}{\partial X_1} - 2X_2(Y_2 + Y_3) \frac{\partial}{\partial X_2} \\ &\quad - 2X_3 Y_3 \frac{\partial}{\partial X_3}. \end{aligned}$$

$$+Y_3)\frac{\partial}{\partial X_2} - 2X_3Y_3\frac{\partial}{\partial X_3}.$$

Next step:

Now let us try to find $D_X^{(3)}$:

For $i > 1$, let us define $_{Y_1}D_{X_i}$ as in what comes in follow:

$$\begin{aligned} _{Y_1}D_{X_i} &:= \{X_i, Y_1^n\} \\ &= X_iY_1^n - q^{-n}Y_1^nX_i \\ &= (1 - q^{-2n})X_iY_1^n \\ &= (1 - e^{2n\hbar})X_iY_1^n \\ &\sim (1 - (1 + 2n\hbar))X_iY_1^n \\ &= -2n\hbar X_iY_1^n \\ &\sim -2nX_iY_1^n = -2X_iY_1\frac{\partial}{\partial Y_1} \\ &= -2Y_1(X_2 + X_3)\frac{\partial}{\partial Y_1}. \end{aligned}$$

For $i > 2$ we have

$$\begin{aligned} _{Y_2}D_{X_i} &:= \{X_i, Y_2^n\} \\ &= X_iY_2^n - q^{-n}Y_2^nX_i \\ &= (1 - q^{-2n})X_iY_2^n \\ &= (1 - e^{2n\hbar})X_iY_2^n \\ &\sim (1 - (1 + 2n\hbar))X_iY_2^n \\ &= -2n\hbar X_iY_2^n \\ &\sim -2nX_iY_2^n \\ &= -2X_iY_2\frac{\partial}{\partial Y_2} \\ &= -2Y_2X_3\frac{\partial}{\partial Y_2}. \end{aligned}$$

For $i > 3$ we have 0.

Let us again have the following definitions

$$_{Y_1}D_2^X := _{Y_1}D_{X_2} = -2Y_1X_2\frac{\partial Y_1^n}{\partial Y_1};$$

$$_{Y_1}D_3^X := _{Y_1}D_{X_3} = -2Y_1X_3\frac{\partial Y_1^n}{\partial Y_1};$$

$$_{Y_2}D_3^X := _{Y_2}D_{X_3} = -2Y_2X_3\frac{\partial Y_1^n}{\partial Y_2};$$

Now let us define

$$_XD_Y^{(3)} := _{Y_1}D_2^X + _{Y_1}D_3^X + _{Y_2}D_3^X = -Y_1(X_2 + X_3)\frac{\partial}{\partial Y_1} - Y_2X_3\frac{\partial}{\partial Y_2};$$

And now as before we have

$$\begin{aligned} _{X_j}D_1^X &:= _{X_j}D_{X_1} \\ &= 4X_1X_j\frac{\partial X_j^n}{\partial X_j}. \\ _{X_1}D_1^X &:= _{X_1}D_{X_1} \\ &= 2X_1^2\frac{\partial X_1^n}{\partial X_1}. \end{aligned}$$

And in a same way we are able to define for $_{X_j}D_2^X$ and $_{X_j}D_3^X$. So let us

define

$$(2.22) \quad \begin{cases} {}_X D_1^X := {}_{X_1} D_1^X + {}_{X_j}^{j<1} D_1^X + {}_{X_j}^{j>1} D_1^X; \\ {}_X D_2^X := {}_{X_2} D_2^X + {}_{X_j}^{j<2} D_2^X + {}_{X_j}^{j>2} D_2^X; \\ {}_X D_3^X := {}_{X_3} D_3^X + {}_{X_j}^{j<3} D_3^X + {}_{X_j}^{j>3} D_3^X; \end{cases}$$

Then we will have

$${}_X D_1^X = X_1^2 \frac{\partial}{\partial X_1} + \sum_{j<1} 2X_1 X_j \frac{\partial}{\partial X_j} + 0$$

And

$${}_X D_2^X = X_2^2 \frac{\partial}{\partial X_2} + \sum_{j<2} 2X_2 X_j \frac{\partial}{\partial X_j} + 0$$

And

$${}_X D_3^X = X_3^2 \frac{\partial}{\partial X_3} + \sum_{j<2} 2X_3 X_j \frac{\partial}{\partial X_j} + 0$$

So we will have

$$\begin{aligned} {}_X D_X^{(3)} &:= {}_X D_1 + {}_X D_2 + {}_X D_3 \\ &= X_1(X_1 + 2X_2 + 2X_3) \frac{\partial}{\partial X_1} + X_2(X_2 + 2X_3) \frac{\partial}{\partial X_2} \\ &\quad + X_3^2 \frac{\partial}{\partial X_3}. \end{aligned}$$

And therefore as in (2.10) we will have the following system of PDEs

$$(2.23) \quad \begin{cases} (X_1(X_1 + 2X_2 + 2X_3) \frac{\partial \tau_1^{(3)}}{\partial X_1} + X_2(X_2 + 2X_3) \frac{\partial \tau_1^{(3)}}{\partial X_2} + X_3^2 \frac{\partial \tau_1^{(3)}}{\partial X_3} \\ - Y_1(X_1 + X_2 + X_3) \frac{\partial \tau_1^{(3)}}{\partial Y_1} - Y_2(X_2 + X_3) \frac{\partial \tau_1^{(3)}}{\partial Y_2} - Y_3 X_3 \frac{\partial \tau_1^{(3)}}{\partial Y_3}) = 0; \\ (2X_1 \frac{\partial \tau_1^{(3)}}{\partial X_1} + 2X_2 \frac{\partial \tau_1^{(3)}}{\partial X_2} + 2X_3 \frac{\partial \tau_1^{(3)}}{\partial X_3} - Y_1 \frac{\partial \tau_1^{(3)}}{\partial Y_1} - Y_2 \frac{\partial \tau_1^{(3)}}{\partial Y_2} - Y_3 \frac{\partial \tau_1^{(3)}}{\partial Y_3}) = 0; \\ D_Y^{(3)} = (Y_1(Y_1 + 2Y_2 + 2Y_3) \frac{\partial \tau_1^{(3)}}{\partial Y_1} + Y_2(Y_2 + 2Y_3) \frac{\partial \tau_1^{(3)}}{\partial Y_2} + Y_3^2 \frac{\partial \tau_1^{(3)}}{\partial Y_3} \\ - Y_1(X_1 + X_2 + X_3) \frac{\partial \tau_1^{(3)}}{\partial Y_1} - Y_2(X_2 + X_3) \frac{\partial \tau_1^{(3)}}{\partial Y_2} - Y_3 X_3 \frac{\partial \tau_1^{(3)}}{\partial Y_3}) = 0; \\ (2X_1 \frac{\partial \tau_1^{(3)}}{\partial X_1} + 2X_2 \frac{\partial \tau_1^{(3)}}{\partial X_2} + 2X_3 \frac{\partial \tau_1^{(3)}}{\partial X_3} - Y_1 \frac{\partial \tau_1^{(3)}}{\partial Y_1} - Y_2 \frac{\partial \tau_1^{(3)}}{\partial Y_2} - Y_3 \frac{\partial \tau_1^{(3)}}{\partial Y_3}) = 0; \end{cases}$$

And according to appendix A we have the following functional dependent nontrivial solution for the whole system of PDEs (2.23)

$$(2.24) \quad \tau_1^{(3)} = \frac{(\sum_{1 \leq i \leq j \leq 2} X_i Y_j)(\sum_{1 \leq i \leq j \leq 2} X_{i+1} Y_{j+1})}{X_2 Y_2 (\sum_{1 \leq i \leq j \leq 3} X_i Y_j)};$$

And again as before, (3) goes back to 3 in the Sl_3 and 1 is a default index which later we will use it for to employ our shifting operators.

According to the number of variables, we will have 6 shifts and then after that it will be in a loop.

So here in sl_3 case we have six solutions which belong to the fraction ring

of polynomial functions.

$$(2.25) \quad \begin{cases} \tau_1^{(3)}[X_1, Y_1, X_2, Y_2, X_3, Y_3] = \frac{X_2 Y_2 (X_3 Y_3 + X_2 (Y_2 + Y_3) + X_1 (Y_1 + Y_2 + Y_3))}{(X_2 Y_2 + X_1 (Y_1 + Y_2)) (X_3 Y_3 + X_2 (Y_2 + Y_3))}; \\ \tau_2^{(3)}[Y_1, X_2, Y_2, X_3, Y_3, X_4] = \frac{X_3 Y_2 (X_2 Y_1 + (X_3 + X_4) (Y_1 + Y_2) + X_4 Y_3)}{(X_2 Y_1 + X_3 (Y_1 + Y_2)) (X_3 Y_2 + X_4 (Y_2 + Y_3))}; \\ \tau_3^{(3)}[X_2, Y_2, X_3, Y_3, X_4, Y_4] = \frac{X_3 Y_3 (X_4 Y_4 + X_3 (Y_3 + Y_4) + X_2 (Y_2 + Y_3 + Y_4))}{(X_3 Y_3 + X_2 (Y_2 + Y_3)) (X_4 Y_4 + X_3 (Y_3 + Y_4))}; \\ \tau_4^{(3)}[Y_2, X_3, Y_3, X_4, Y_4, X_5] = \frac{X_4 Y_3 (X_3 Y_2 + (X_4 + X_5) (Y_2 + Y_3) + X_5 Y_4)}{(X_3 Y_2 + X_4 (Y_2 + Y_3)) (X_4 Y_3 + X_5 (Y_3 + Y_4))}; \\ \tau_5^{(3)}[X_3, Y_3, X_4, Y_4, X_5, Y_5] = \frac{X_4 Y_4 (X_5 Y_5 + X_4 (Y_4 + Y_5) + X_3 (Y_3 + Y_4 + Y_5))}{(X_4 Y_4 + X_3 (Y_3 + Y_4)) (X_5 Y_5 + X_4 (Y_4 + Y_5))}; \\ \tau_6^{(3)}[Y_3, X_4, Y_4, X_5, Y_5, X_6] = \frac{X_5 Y_4 (X_4 Y_3 + (X_5 + X_6) (Y_3 + Y_4) + X_6 Y_5)}{(X_4 Y_3 + X_5 (Y_3 + Y_4)) (X_5 Y_4 + X_6 (Y_4 + Y_5))}; \end{cases}$$

Where $\tau_1^{(3)} := \tau_1^{(3)}[\dots, X_1, Y_1, X_2, Y_2, X_3, Y_3 \dots]$.

Again by setting $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and according to (2.4) we have to write down the following brackets as a composition of $\tau_i^{(3)}$ s, because of algebra structure and it will be done by using Mathematica coding in appendix A.

$$(2.26) \quad \begin{cases} F_2^{(3)} = \{\tau_1^{(3)}, \tau_2^{(3)}\} = -(1 - \tau_1^{(3)})(1 - \tau_2^{(3)})(\tau_1^{(3)} \tau_2^{(3)}); \\ F_3^{(3)} = \{\tau_1^{(3)}, \tau_3^{(3)}\} = (1 - \tau_1^{(3)})(1 - \tau_3^{(3)})(\tau_1^{(3)} \tau_2^{(3)} + \tau_2^{(3)} \tau_3^{(3)} - \tau_2^{(3)}); \\ F_4^{(3)} = \{\tau_1^{(3)}, \tau_4^{(3)}\} = -(1 - \tau_1^{(3)})(1 - \tau_4^{(3)}) \\ \quad (\tau_1^{(3)} \tau_2^{(3)} + \tau_2^{(3)} \tau_3^{(3)} + \tau_3^{(3)} \tau_4^{(3)} - \tau_1^{(3)} - \tau_2^{(3)} - \tau_3^{(3)} - \tau_4^{(3)} + 1); \\ F_5^{(3)} = \{\tau_1^{(3)}, \tau_5^{(3)}\} = (1 - \tau_1^{(3)})(1 - \tau_5^{(3)})(\tau_2^{(3)} \tau_3^{(3)} + \tau_3^{(3)} \tau_4^{(3)} - \tau_2^{(3)} \\ \quad - \tau_3^{(3)} - \tau_4^{(3)} + 1); \\ F_6^{(3)} = \{\tau_1^{(3)}, \tau_6^{(3)}\} = -(1 - \tau_1^{(3)})(1 - \tau_6^{(3)})(\tau_3^{(3)} \tau_4^{(3)} - \tau_4^{(3)} - \tau_3^{(3)} + 1); \\ F_i^{(3)} = \{\tau_1^{(3)}, \tau_i^{(3)}\} = 0 \quad \text{for } |i - 1| \geq 6; \end{cases}$$

2.3. Lattice W_4 algebra; main generator. In this case we will use the following defined Poisson bracket based on Cartan matrix

$$A_3 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

But for to do this according to our previous ordering and list of variables, and the same as what we din in sl_3 case, let us for simplicity set our set of variables as follows:

Set $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and so on.

Definition 2.4. Let's define our Poisson bracket as follows in the case of sl_3 :

$$(2.27) \quad \left\{ \begin{array}{ll} \{X_i, X_j\} := 2X_iX_j & \text{if } i < j; \\ \{Y_i, Y_j\} := 2Y_iY_j & \text{if } i < j; \\ \{Z_i, Z_j\} := 2Z_iZ_j & \text{if } i < j; \\ \{X_i, X_i\} := 0; \\ \{Y_i, Y_i\} := 0; \\ \{Z_i, Z_i\} := 0; \\ \{X_i, Y_j\} := X_iY_j & \text{if } i > j; \\ \{X_i, Y_j\} := -X_iY_j & \text{if } i \leq j; \\ \{X_i, Z_j\} := 0; \\ \{Y_i, Z_j\} := Y_iZ_j & \text{if } i > j; \\ \{Y_i, Z_j\} := -Y_iZ_j & \text{if } i \leq j; \end{array} \right.$$

As it comes out that, here our set of variables will be as follows:

$$\begin{array}{cccccccccccccccccccc} \circ & \cdots & \circ & \circ & \circ & \circ & \circ & \cdots & \circ & \circ & \circ & \circ & \circ & \circ & \cdots & \circ \\ -\infty & & X_1 & Y_1 & Z_1 & X_2 & Y_2 & Z_2 & & X_{n-1} & Y_{n-1} & Z_{n-1} & X_n & Y_n & Z_n & +\infty \end{array}$$

And instead of (2.1) we will have the following q -commutation relations for $j \in \{1, 2, 3\}$ and as always $i \in \{1, 2, 3\}$:

$$(2.28) \quad \left\{ \begin{array}{ll} X_iX_j = q^2X_jX_i & \text{if } i \leq j \\ Y_iY_j = q^2Y_jY_i & \text{if } i \leq j \\ Z_iZ_j = q^2Z_jZ_i & \text{if } i \leq j \\ X_iY_j = q^{-1}Y_jX_i & \text{if } i \leq j \\ Y_iZ_j = q^{-1}Z_jY_i & \text{if } i \leq j \\ X_iZ_j = Z_jX_i & \end{array} \right.$$

And by using the same approach as what we did for sl_2 and sl_3 , it became clear that the equations $D_X^{(4)}$, $D_Y^{(4)}$ and $D_Z^{(4)}$ and also $H_X^{(4)}$, $H_Y^{(4)}$ and $H_Z^{(4)}$ will have the following forms:

(2.29)

$$\mathfrak{D}_X^{(4)} = X_1(X_1+2X_2+2X_3)\frac{\partial\tau_1^{(4)}}{\partial X_1} + X_2(X_2+2X_3)\frac{\partial\tau_1^{(4)}}{\partial X_2} + X_3^2\frac{\partial\tau_1^{(4)}}{\partial X_3} - Y_1(X_2+X_3)$$

$$\frac{\partial\tau_1^{(4)}}{\partial Y_1} - Y_2X_3\frac{\partial\tau_1^{(4)}}{\partial Y_2};$$

(2.30)

$$\mathfrak{D}_Y^{(4)} = Y_1(Y_1+2Y_2+2Y_3)\frac{\partial\tau_1^{(4)}}{\partial Y_1} + Y_2(Y_2+2Y_3)\frac{\partial\tau_1^{(4)}}{\partial Y_2} + Y_3^2\frac{\partial\tau_1^{(4)}}{\partial Y_3} - X_1(Y_1+Y_2+Y_3)$$

$$\frac{\partial\tau_1^{(4)}}{\partial X_1} - X_2(Y_2+Y_3)\frac{\partial\tau_1^{(4)}}{\partial X_2} - X_3Y_3\frac{\partial\tau_1^{(4)}}{\partial X_3} - Z_1(Y_2+Y_3)\frac{\partial\tau_1^{(4)}}{\partial Z_1} - Z_2Y_3\frac{\partial\tau_1^{(4)}}{\partial Z_2};$$

(2.31)

$$\mathfrak{D}_Z^{(4)} = Z_1(Z_1+2Z_2+2Z_3)\frac{\partial\tau_1^{(4)}}{\partial Z_1} + Z_2(Z_2+2Z_3)\frac{\partial\tau_1^{(4)}}{\partial Z_2} + Z_3^2\frac{\partial\tau_1^{(4)}}{\partial Z_3} - Y_1(Z_1+Z_2+Z_3)$$

$$\frac{\partial \tau_1^{(4)}}{\partial Y_1} - Y_2(Z_2 + Z_3) \frac{\partial \tau_1^{(4)}}{\partial Y_2} - Y_3 Z_3 \frac{\partial \tau_1^{(4)}}{\partial Y_3};$$

(2.32)

$$H_X^{(4)} = 2X_1 \frac{\partial \tau_1^{(4)}}{\partial X_1} + 2X_2 \frac{\partial \tau_1^{(4)}}{\partial X_2} + 2X_3 \frac{\partial \tau_1^{(4)}}{\partial X_3} - Y_1 \frac{\partial \tau_1^{(4)}}{\partial Y_1} - Y_2 \frac{\partial \tau_1^{(4)}}{\partial Y_2} - Y_3 \frac{\partial \tau_1^{(4)}}{\partial Y_3};$$

(2.33)

$$H_Y^{(4)} = 2Y_1 \frac{\partial \tau_1^{(4)}}{\partial Y_1} + 2Y_2 \frac{\partial \tau_1^{(4)}}{\partial Y_2} + 2Y_3 \frac{\partial \tau_1^{(4)}}{\partial Y_3} - X_1 \frac{\partial \tau_1^{(4)}}{\partial X_1} - X_2 \frac{\partial \tau_1^{(4)}}{\partial X_2} - X_3 \frac{\partial \tau_1^{(4)}}{\partial X_3} - z_1 \frac{\partial \tau_1^{(4)}}{\partial Z_1} - Z_2 \frac{\partial \tau_1^{(4)}}{\partial Z_2} - Z_3 \frac{\partial \tau_1^{(4)}}{\partial Z_3};$$

$$(2.34) \quad H_Z^{(4)} = 2Z_1 \frac{\partial \tau_1^{(4)}}{\partial Z_1} + 2Z_2 \frac{\partial \tau_1^{(4)}}{\partial Z_2} + 2Z_3 \frac{\partial \tau_1^{(4)}}{\partial Z_3} - Y_1 \frac{\partial \tau_1^{(4)}}{\partial Y_1} - Y_2 \frac{\partial \tau_1^{(4)}}{\partial Y_2} - Y_3 \frac{\partial \tau_1^{(4)}}{\partial Y_3};$$

And the functional dependent nontrivial solutions for the whole system of first order partial differential equation is as follows:

$$(2.35) \quad \tau_1^{(4)} = \frac{(\sum_{1 \leq i \leq j \leq m \leq 2} x_i y_j z_m)(\sum_{1 \leq i \leq j \leq m \leq 2} x_{i+1} y_{j+1} z_{m+1})}{x_2 y_2 z_2 (\sum_{1 \leq i \leq j \leq m \leq 3} x_i y_j z_m)};$$

And again as before, (4) goes back to 4 in the Sl_4 and 1 is a default index which later we will use it for to employ our shifting operators.

According to the number of variables, we will have 9 shifts and then after that it will be in a loop.

So here in sl_4 case we have nine solutions:

$$\tau_1^{(4)} := \tau_1^{(4)}[X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_3, Y_3, Z_3];$$

$$\tau_2^{(4)} := \tau_1^{(4)}[X_1 \rightarrow Y_1, Y_1 \rightarrow Z_1, Z_1 \rightarrow X_2, X_2 \rightarrow Y_2, Y_2 \rightarrow Z_2, Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3];$$

$$\tau_3^{(4)} := \tau_2^{(4)}[Y_1 \rightarrow Z_1, Z_1 \rightarrow X_2, X_2 \rightarrow Y_2, Y_2 \rightarrow Z_2, Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4];$$

$$\tau_4^{(4)} := \tau_3^{(4)}[Z_1 \rightarrow X_2, X_2 \rightarrow Y_2, Y_2 \rightarrow Z_2, Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4];$$

$$\tau_5^{(4)} := \tau_4^{(4)}[X_2 \rightarrow Y_2, Y_2 \rightarrow Z_2, Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4, Y_4 \rightarrow Z_4];$$

$$\tau_6^{(4)} := \tau_5^{(4)}[Y_2 \rightarrow Z_2, Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4, Y_4 \rightarrow Z_4, Z_4 \rightarrow X_5];$$

$$\tau_7^{(4)} := \tau_6^{(4)} [Z_2 \rightarrow X_3, X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4, Y_4 \rightarrow Z_4, Z_4 \rightarrow X_5, X_5 \rightarrow Y_5];$$

$$\tau_8^{(4)} := \tau_7^{(4)} [X_3 \rightarrow Y_3, Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4, Y_4 \rightarrow Z_4, Z_4 \rightarrow X_5, X_5 \rightarrow Y_5, Y_5 \rightarrow Z_5];$$

$$\tau_9^{(4)} := \tau_8^{(4)} [Y_3 \rightarrow Z_3, Z_3 \rightarrow X_4, X_4 \rightarrow Y_4, Y_4 \rightarrow Z_4, Z_4 \rightarrow X_5, X_5 \rightarrow Y_5, Y_5 \rightarrow Z_5, Z_5 \rightarrow X_6];$$

which belong to the fraction ring of polynomial functions.

2.4. Lattice W_5 algebra; main generator. In this case we will use the following defined Poisson bracket based on Cartan matrix

$$A_4 = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

But for to do this according to our previous ordering and list of variables, and the same as what we din in sl_4 case, let us for simplicity set our set of variables as follows:

Set $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and $X_i^{(4i)} := K_i$ and so on.

Definition 2.5. Let's define our Poisson bracket as follows in the case of sl_4 :

$$(2.36) \quad \left\{ \begin{array}{ll} \{X_i, X_j\} := 2X_i X_j & \text{if } i < j; \\ \{Y_i, Y_j\} := 2Y_i Y_j & \text{if } i < j; \\ \{Z_i, Z_j\} := 2Z_i Z_j & \text{if } i < j; \\ \{K_i, K_j\} := 2K_i K_j & \text{if } i < j; \\ \{X_i, X_i\} := 0; \\ \{Y_i, Y_i\} := 0; \\ \{Z_i, Z_i\} := 0; \\ \{K_i, K_i\} := 0; \\ \{X_i, Y_j\} := X_i Y_j & \text{if } i > j; \\ \{X_i, Y_j\} := -X_i Y_j & \text{if } i \leq j; \\ \{X_i, Z_j\} := 0; \\ \{X_i, K_j\} := 0; \\ \{Y_i, Z_j\} := Y_i Z_j & \text{if } i > j; \\ \{Y_i, Z_j\} := -Y_i Z_j & \text{if } i \leq j; \\ \{Y_i, K_j\} := Y_i K_j & \text{if } i > j; \\ \{Y_i, K_j\} := -Y_i K_j & \text{if } i \leq j; \end{array} \right.$$

As it comes out that, here our set of variables will be as follows:

$$\overset{\circ}{-}\infty \cdots \overset{\circ}{-}X_1 \overset{\circ}{-}Y_1 \overset{\circ}{-}Z_1 \overset{\circ}{-}K_1 \overset{\circ}{-}X_2 \overset{\circ}{-}Y_2 \overset{\circ}{-}Z_2 \overset{\circ}{-}K_2 \cdots \overset{\circ}{-}X_n \overset{\circ}{-}Y_n \overset{\circ}{-}Z_n \overset{\circ}{-}K_n \cdots \overset{\circ}{-}\infty$$

And instead of (2.1) we will have the following q -commutation relations for $j \in \{1, 2, 3\}$ and as always $i \in \{1, 2, 3\}$:

$$\begin{cases} x_i x_j = q^2 x_j x_i & \text{if } i \leq j \\ y_i y_j = q^2 y_j y_i & \text{if } i \leq j \\ z_i z_j = q^2 z_j z_i & \text{if } i \leq j \\ k_i k_j = q^2 k_j k_i & \text{if } i \leq j \\ x_i y_j = q^{-1} y_j x_i & \text{if } i \leq j \\ y_i z_j = q^{-1} z_j y_i & \text{if } i \leq j \\ z_i k_j = q^{-1} k_j z_i & \text{if } i \leq j \\ x_i z_j = z_j x_i \\ y_i k_j = k_j y_i \\ x_i k_j = k_j x_i \end{cases}$$

And by using the same approach as what we did for sl_2 and sl_3 and sl_4 , it became clear that the equations $D_X^{(5)}$, $D_Y^{(5)}$, $D_Z^{(5)}$ and $D_K^{(5)}$ and also $H_X^{(5)}$, $H_Y^{(5)}$, $H_Z^{(5)}$ and $H_K^{(5)}$ will have the following forms:

$$\mathfrak{D}_X^{(5)} = X_1(X_1+2X_2+2X_3)\frac{\partial\tau_1^{(5)}}{\partial X_1} + X_2(X_2+2X_3)\frac{\partial\tau_1^{(5)}}{\partial X_2} + X_3^2\frac{\partial\tau_1^{(5)}}{\partial X_3} - Y_1(X_2+X_3)$$

$$\frac{\partial\tau_1^{(5)}}{\partial Y_1} - Y_2X_3\frac{\partial\tau_1^{(5)}}{\partial Y_2};$$

(2.38)

$$\mathfrak{D}_Y^{(5)} = Y_1(Y_1+2Y_2+2Y_3)\frac{\partial\tau_1^{(5)}}{\partial Y_1} + Y_2(Y_2+2Y_3)\frac{\partial\tau_1^{(5)}}{\partial Y_2} + Y_3^2\frac{\partial\tau_1^{(5)}}{\partial Y_3} - X_1(Y_1+Y_2+Y_3)$$

$$\frac{\partial\tau_1^{(5)}}{\partial X_1} - X_2(Y_2+Y_3)\frac{\partial\tau_1^{(5)}}{\partial X_2} - X_3Y_3\frac{\partial\tau_1^{(5)}}{\partial X_3} - Z_1(Y_2+Y_3)\frac{\partial\tau_1^{(5)}}{\partial z_1} - Z_2Y_3\frac{\partial\tau_1^{(5)}}{\partial Z_2};$$

(2.39)

$$\mathfrak{D}_Z^{(5)} = Z_1(Z_1+2Z_2+2Z_3)\frac{\partial\tau_1^{(5)}}{\partial Z_1} + Z_2(Z_2+2Z_3)\frac{\partial\tau_1^{(5)}}{\partial Z_2} + Z_3^2\frac{\partial\tau_1^{(5)}}{\partial Z_3} - Y_1(Z_1+Z_2+Z_3)$$

$$\frac{\partial\tau_1^{(5)}}{\partial Y_1} - Y_2(Z_2+Z_3)\frac{\partial\tau_1^{(5)}}{\partial Y_2} - Y_3Z_3\frac{\partial\tau_1^{(5)}}{\partial Y_3} - K_1(Z_2+Z_3)\frac{\partial\tau_1^{(5)}}{\partial k_1} - K_2Z_3\frac{\partial\tau_1^{(5)}}{\partial K_2};$$

(2.40)

$$\mathfrak{D}_K^{(5)} = K_1(K_1+2K_2+2K_3)\frac{\partial\tau_1^{(5)}}{\partial K_1} + K_2(K_2+2K_3)\frac{\partial\tau_1^{(5)}}{\partial K_2} + K_3^2\frac{\partial\tau_1^{(5)}}{\partial Z_3} - Z_1(K_1+K_2$$

$$+K_3)\frac{\partial\tau_1^{(5)}}{\partial Z_1} - Z_2(K_2+K_3)\frac{\partial\tau_1^{(5)}}{\partial z_2} - Z_3X_3\frac{\partial\tau_1^{(5)}}{\partial Z_3};$$

(2.41)

$$H_X^{(5)} = 2X_1\frac{\partial\tau_1^{(5)}}{\partial X_1} + 2X_2\frac{\partial\tau_1^{(5)}}{\partial X_2} + 2X_3\frac{\partial\tau_1^{(5)}}{\partial X_3} - Y_1\frac{\partial\tau_1^{(5)}}{\partial Y_1} - Y_2\frac{\partial\tau_1^{(5)}}{\partial Y_2} - Y_3\frac{\partial\tau_1^{(5)}}{\partial Y_3};$$

(2.42)

$$H_Y^{(5)} = 2Y_1 \frac{\partial \tau_1^{(5)}}{\partial Y_1} + 2Y_2 \frac{\partial \tau_1^{(5)}}{\partial Y_2} + 2Y_3 \frac{\partial \tau_1^{(5)}}{\partial Y_3} - X_1 \frac{\partial \tau_1^{(5)}}{\partial X_1} - X_2 \frac{\partial \tau_1^{(5)}}{\partial X_2} - X_3 \frac{\partial \tau_1^{(5)}}{\partial X_3} - Z_1 \frac{\partial \tau_1^{(5)}}{\partial Z_1} - Z_2 \frac{\partial \tau_1^{(5)}}{\partial Z_2} - Z_3 \frac{\partial \tau_1^{(5)}}{\partial Z_3};$$

(2.43)

$$H_Z^{(5)} = 2Z_1 \frac{\partial \tau_1^{(5)}}{\partial Z_1} + 2Z_2 \frac{\partial \tau_1^{(5)}}{\partial Z_2} + 2Z_3 \frac{\partial \tau_1^{(5)}}{\partial Z_3} - Y_1 \frac{\partial \tau_1^{(5)}}{\partial Y_1} - Y_2 \frac{\partial \tau_1^{(5)}}{\partial Y_2} - Y_3 \frac{\partial \tau_1^{(5)}}{\partial Y_3} - K_1 \frac{\partial \tau_1^{(5)}}{\partial K_1} - K_2 \frac{\partial \tau_1^{(5)}}{\partial K_2} - K_3 \frac{\partial \tau_1^{(5)}}{\partial K_3};$$

(2.44)

$$H_K^{(5)} = 2K_1 \frac{\partial \tau_1^{(5)}}{\partial K_1} + 2K_2 \frac{\partial \tau_1^{(5)}}{\partial K_2} + 2K_3 \frac{\partial \tau_1^{(5)}}{\partial K_3} - Z_1 \frac{\partial \tau_1^{(5)}}{\partial Z_1} - Z_2 \frac{\partial \tau_1^{(5)}}{\partial Z_2} - Z_3 \frac{\partial \tau_1^{(5)}}{\partial Z_3};$$

And the functional dependent nontrivial solutions for the whole system of first order partial differential equation is as follows:

$$(2.45) \quad \tau_1^{(5)} = \frac{(\sum_{1 \leq i \leq j \leq m \leq l \leq 2} x_i y_j z_m k_l)(\sum_{1 \leq i \leq j \leq m \leq l \leq 2} x_{i+1} y_{j+1} z_{m+1} k_{l+1})}{x_2 y_2 z_2 k_2 (\sum_{1 \leq i \leq j \leq m \leq l \leq 3} x_i y_j z_m k_l)};$$

And again as before, (5) goes back to 5 in the Sl_5 and 1 is a default index which later we will use it for to employ our shifting operators.

According to the number of variables, we will have 12 shifts and then after that it will be in a loop.

So here in sl_5 case we have twelve solutions just as what we did in sl_4 , and here skip to write them down.

2.5. Lattice W_n algebra; main generator. Here for sl_n , we skip to write down all steps which we have done in previous sections and just will write down our main generator of the lattice W_n algebra.

The functional dependent nontrivial solution for the whole system of first order partial differential equations will be as what comes in follow:

$$(2.46) \quad \tau_1^{(n)} = \frac{(\sum_{1 \leq i_1 \leq i_2 \dots \leq i_{n-1} \leq 2} x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_{n-1}}^{(n-1)}) (\sum_{1 \leq i_1 \leq i_2 \dots \leq i_{n-1} \leq 2} x_{i_1+1}^{(1)} x_{i_2+1}^{(2)} \dots x_{i_{n-1}+1}^{(n-1)})}{x_2^{(1)} \dots x_2^{(n-1)} (\sum_{1 \leq i_1 \leq i_2 \dots \leq i_{n-1} \leq 3} x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_{n-1}}^{(n-1)})};$$

We should notice that $x_{i_j}^{(j)}$ s are different of each other for any $j \in \{1, \dots, n-1\}$

ACKNOWLEDGEMENT

The research in this article would have taken far longer to complete without the encouragement from many others. It is a delight to acknowledge those who have supported me over the last three years.

I would like to thank my supervisor, Prof. Yaroslav Pugai, for his guidance and relaxed, thoughtful insight.

I thank all of Institute for information transmission problems (Kharkevich institute)'s staff for their hospitality over the last one year.

I am particularly thankful for the help and advice of Prof. Brendan Godfrey, without whom the learning Mathematica would have been very much steeper and unimaginable. And finally I would like to thank Professor Boris Feigin for to suggesting me this interesting problem and enlightening discussions during the preparation for my first article which was my first step in this subject!

REFERENCES

- [1] Razavinia, Farrokh. "Local coordinate systems on quantum flag manifolds." arXiv preprint arXiv:1610.09443 (2016).
- [2] Pugay, Ya P. "Lattice W algebras and quantum groups." Theoretical and Mathematical Physics 100.1 (1994): 900-911.
- [3] Goursat, Edouard. *A Course in Mathematical Analysis: pt. 2. Differential equations.* [c1917. Vol. 2. Dover Publications, 1916.
- [4] Caressa, Paolo. "The algebra of Poisson brackets." Young Algebra Seminar, Roma Tor Vergata. 2000.
- [5] Hong, Jin, and Seok-Jin Kang. *Introduction to quantum groups and crystal bases.* Vol. 42. American Mathematical Soc., 2002.
- [6] Feigin, B.L.: talk at RIMS 1992.
- [7] Hikami, Kazuhiro, and Rei Inoue. "Classical lattice W algebras and integrable systems." Journal of Physics A: Mathematical and General 30.19 (1997): 6911.
- [8] Hikami, Kazuhiro. "Lattice WN algebra and its quantization." Nuclear Physics B 505.3 (1997): 749-770.
- [9] Antonov, Alexander, Alexander A. Belov, and Karén Chaltikian. "Lattice conformal theories and their integrable perturbations." Journal of Geometry and Physics 22.4 (1997): 298-318.
- [10] Belov, Alexander A., and Karen D. Chaltikian. "Lattice analogues of W-algebras and classical integrable equations." Physics Letters B 309.3-4 (1993): 268-274.
- [11] Berenstein, Arkady. "Group-like elements in quantum groups, and Feigin's conjecture." arXiv preprint q-alg/9605016 (1996).

3. APPENDIX A

Brendan B. Godfrey and Farrokh Razavinia
brendan.godfrey@ieee.org
f.razavinia@phystech.edu

This section has been completed by getting help from professor Brendan B. Godfrey from Institute for Research in Electronics and Applied Physics (The University of Maryland), in a direct communications and discussions through email and also through a series of questions and discussions in mathematica stackexchange.

And I have to say that without his great Mathematica skills, it nearly was impossible to get such an interesting results!

In this appendix you will be able to see some parts of Mathematica codings which we have used for to obtain our algebra structures.

And we believe that what is written in this appendix can open a new approach in solving the following system of q -linear homogeneous equations in one unknown f .

$$(3.1) \quad \begin{cases} equ_1(f) = a_{11} \frac{\partial f}{\partial x_1} + a_{21} \frac{\partial f}{\partial x_2} + \cdots + a_{n1} \frac{\partial f}{\partial x_n} = 0 \\ equ_2(f) = a_{12} \frac{\partial f}{\partial x_1} + a_{22} \frac{\partial f}{\partial x_2} + \cdots + a_{n2} \frac{\partial f}{\partial x_n} = 0 \\ \vdots \\ equ_q(f) = a_{1q} \frac{\partial f}{\partial x_1} + a_{2q} \frac{\partial f}{\partial x_2} + \cdots + a_{nq} \frac{\partial f}{\partial x_n} = 0 \end{cases}$$

Where the coefficients a_{ik} are functions of n independent variables x_1, \dots, x_n and do not contain the unknown function f . [3]

And we have to mention that, to reach to this point was impossible without using Mathematica!

3.1. Lattice W_3 algebra. .

LISTING 1. Example code

```

1 p = D[f[x1, x2, x3, y1, y2, y3], x1];
2 q = D[f[x1, x2, x3, y1, y2, y3], x2];
3 r = D[f[x1, x2, x3, y1, y2, y3], x3];
4 o = D[f[x1, x2, x3, y1, y2, y3], y1];
5 x = D[f[x1, x2, x3, y1, y2, y3], y2];
6 a = D[f[x1, x2, x3, y1, y2, y3], y3];
7 equ1 = 2 x1 p + 2 x2 q + 2 x3 r - y1 o - y2 x - y3 a;
8 equ2 = -x1 p - x2 q - x3 r + 2 y1 o + 2 y2 x + 2 y3 a;
9 equ3 = (x1 (x1 + 2 x2 + 2 x3)) p + (x2 (x2 + 2 x3)) q + (x3^2) r
10 - (y1 (x2 + x3)) o - y2 x3 x;
11 equ4 = (y1 (y1 + 2 y2 + 2 y3)) o + (y2 (y2 + 2 y3)) x + (y3^2) a
12 - (x1 (y1 + y2 + y3)) p - x2 (y2 + y3) q - x3 y3 r;
13 DSolve[{equ1 == 0, equ2 == 0, equ3 == 0, equ4 == 0}, f, {x1, x2, x3, y1, y2, y3}]

```

As you see *DSolve* returns un-evaluated i.e. it means that it is not able to solve our system of first order partial differential equations.

LISTING 2. Example code

```

1 DSolve[2 equ1 + equ2 == 0, f[x1, x2, x3, y1, y2, y3], {x1, x2, x3, y1, y2, y3}][[1, 1]]
2 (* f[x1, x2, x3, y1, y2, y3] -> C[1][x2/x1, x3/x1, y1, y2, y3] *)
3 DSolve[equ1 + 2 equ2 == 0, f[x1, x2, x3, y1, y2, y3], {x1, x2, x3, y1, y2, y3}][[1, 1]]
4 (* f[x1, x2, x3, y1, y2, y3] -> C[1][x1, x2, x3, y2/y1, y3/y1] *)

```

Consequently, the dimensionality of this problem can be reduced from six to four.

LISTING 3. Example code

```

1 f[x1_, x2_, x3_, y1_, y2_, y3_] := g[x2/x1, x3/x1, y2/y1, y3/y1]
2 equ5 = FullSimplify[(equ3/x1) /. {x2 -> v2 x1, x3 -> v3 x1, y2 -> w2 y1, y3 -> w3 y1}
3 ]
4 (* (v2 + v3)*w3*Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] +
5 v2*w2*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] -
6 v3*(1 + 2*v2 + v3)*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] -
7 v2*(1 + v2)*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)
8 equ6 = FullSimplify[(equ4/y1) /. {x2 -> v2 x1, x3 -> v3 x1, y2 -> w2 y1, y3 -> w3 y1}
9 ]
10 (* -(w3*(1 + 2*w2 + w3)*Derivative[0, 0, 0, 1][g][v2, v3, w2, w3]) -
11 w2*(1 + w2)*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] +
12 v3*(1 + w2)*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] +
13 v2*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)

```

Although *DSolve* cannot solve these equations as a pair either. But it can solve each separately.

LISTING 4. Example code

```
1 DSolve[equ5 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]] /. C[1] -> c5
2 (* g[v2, v3, w2, w3] -> c5[(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2] *)
3 (DSolve[equ6 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]] /.
4 C[1] -> c6) // FullSimplify
5 (* g[v2, v3, w2, w3] -> c6[-((v3 (1 + w2))/v2), ((1 + w2) (1 + w2 + w3))/(v2 w3),
6 -Log[(1 + w2)/(v2 w2)]] *)
```

The first results indicates that g is a function of

LISTING 5. Example code

```
1 var5 = List @@ %%%[[2]]
2 (* {(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2} *)
```

and also

LISTING 6. Example code

```
1 var6 = List @@ %%%[[2]]
2 (* {-(v3 (1 + w2))/v2, ((1 + w2) (1 + w2 + w3))/(v2 w3), -Log[(1 + w2)/(v2 w2)]] *)
```

The second list of functions can be simplified by

LISTING 7. Example code

```
1 var6 [[3]] = Exp[var6 [[3]]];
2 var6 [[1]] = -var6 [[1]] var6 [[3]];
3 var6 [[2]] = var6 [[2]] var6 [[3]];
4 var6
5 (* {v3 w2, (w2 (1 + w2 + w3))/w3, (v2 w2)/(1 + w2)} *)
```

Now the next step is to combine the previous two expressions for g for to obtain a single expression, presumably as a function of two variables.

The system of PDEs above can be solved using the procedure described in Chapter V, Sec IV of Goursat's Differential Equations [3].

The first step is to find the complete, non-commutative group of differential operators that includes *equ5* and *equ6*.

LISTING 8. Example code

```
1 comm[equa_, equb_] :=
2 Collect[(equa /. {Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] -> D[equb, v2],
3 Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] -> D[equb, v3],
4 Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] -> D[equb, w2],
5 Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] -> D[equb, w3]}) -
6 (equb /. {Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] -> D[equa, v2],
7 Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] -> D[equa, v3],
8 Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] -> D[equa, w2],
9 Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] -> D[equa, w3]})],
10 {Derivative[1, 0, 0, 0][g][v2, v3, w2, w3], Derivative[0, 1, 0, 0][g][v2, v3, w2, w3],
11 Derivative[0, 0, 1, 0][g][v2, v3, w2, w3], Derivative[0, 0, 0, 1][g][v2, v3, w2, w3]},
12 Simplify]
13 equ7 = comm[equ5, equ6]
14 (* -(w3*(v3*(1 + w2 + w3) + v2*(1 + 2*w2 + w3))*Derivative[0, 0, 0, 1][g][v2, v3, w2,
15 w3])
16 - v2*w2*(1 + w2)*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] + v3*(v3*(1 + w2)
```

```

16 + v2*(2 + w2))*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] +
17 v2^2*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)

```

which by inspection is independent of *equ5* and *equ6*. On the other hand, *comm[equ5, equ7]* and *comm[equ6, equ7]* do not yield independent equations, again by inspection. Thus $\{equ5, equ6, equ7\}$ is a complete group of three operators in four independent variables. From this information alone, we know that *g* is an arbitrary function of precisely one first integral. This first integral can be obtained by systematically eliminating variables and equations, one pair at a time, until a single equation of two variable remains. We start by solving any one of the equations.

LISTING 9. Example code

```

1 DSolve[equ5 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]]
2 (* g[v2, v3, w2, w3] -> C[1]((v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2) *)

```

and use the solution as the basis for a change of variables:

LISTING 10. Example code

```

1 g[v2_, v3_, w2_, w3_] := h[w2, (v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2]
2 solw2 = equ5/(v2 w2) // Simplify
3 (* Derivative[1, 0, 0, 0][h][w2, (v2*(1 + v2 + v3))/v3, (1 + v2)*w2, (v3*w3)/v2] *)

```

indicating that *h* is independent of *w2*. This leaves us with two equations in three variables

LISTING 11. Example code

```

1 newvar = Solve[Thread[{b1, b2, b3, b4} == List @@ solw2], {v2, v3, w2, w3}] // Flatten;
2 (((b3 equ7/(b1 - b3)) // FullSimplify) /. solw2 -> 0 /. newvar) // FullSimplify; \
3 Collect[({equ6 // FullSimplify) /. solw2 -> 0 /. newvar) //
4 FullSimplify, b1, FullSimplify] + % b1/b3
5 equ10 = %% /. Derivative[0, n1_, n2_, n3_][h][b1, b2, b3, b4] ->
6 Derivative[n1, n2, n3][h][b2, b3, b4]
7 (* b4*(b3 + b4 + b2*b4)*Derivative[0, 0, 1][h][b2, b3, b4] + (1 + b3)*
8 (b3*Derivative[0, 1, 0][h][b2, b3, b4] + (1 + b2)*Derivative[1, 0, 0][h][b2, b3, b4]) *)
9 equ11 = %% /. Derivative[0, n1_, n2_, n3_][h][b1, b2, b3, b4] ->
10 Derivative[n1, n2, n3][h][b2, b3, b4]
11 (* (-1 + b4)*b4*Derivative[0, 0, 1][h][b2, b3, b4] + (1 + b2 + b3)
12 *Derivative[1, 0, 0][h][b2, b3, b4] *)

```

Proceeding as before, we next solve one of *equ10* and *equ11*. (We choose the simpler one.)

LISTING 12. Example code

```

1 DSolve[equ11 == 0, h[b2, b3, b4], {b2, b3, b4}][[1, 1, 2]]
2 (* h[b2, b3, b4] -> C[1][b3][Log[(1 - b4)/((1 + b2 + b3) b4)] *)

```

and use it as the basis for a further change of variables.

LISTING 13. Example code

```

1 h[b2_, b3_, b4_] := k[b2, b3, (1 - b4)/((1 + b2 + b3) b4)]
2 solb2 = (equ11/(1 + b2 + b3)) // Simplify
3 (* Derivative[1, 0, 0][k][b2, b3, (1 - b4)/(b4 + b2*b4 + b3*b4)] *)

```

indicating that k is independent of b_2 . This leaves us with one equation in two variables.

LISTING 14. Example code

```

1 newvar1 = Solve[Thread[{c2, c3, c4} == List @@ solb2], {b2, b3, b4}] // Flatten;
2 ((equ10 // FullSimplify) /. solb2 -> 0 /. newvar1) // FullSimplify
3 equ12 = % /. Derivative[0, n1_, n2_][k][c2, c3, c4] -> Derivative[n1, n2][k][c3, c4]
4 (* -((1 + c4 + 2*c3*c4)*Derivative[0, 1][k][c3, c4]) + c3*(1 + c3)
5 *Derivative[1, 0][k][c3, c4] *)

```

Finally, *DSolve* yields

LISTING 15. Example code

```

1 DSolve[equ12 == 0, k[c3, c4], {c3, c4}][[1, 1, 2]]
2 (* k[c3, c4] -> C[1][c3 (1 + c4 + c3 c4)] *)

```

Transforming back to the original independent variables gives

LISTING 16. Example code

```

1 (((% /. Thread[{c2, c3, c4} -> List @@ solb2]) // Simplify) /.
2 Thread[{b1, b2, b3, b4} -> List @@ solw2]) // Simplify
3 (* C[1][((v2 w2 (1 + (1 + v2) w2 + (1 + v2 + v3) w3)))/((v2 + v3 + v3 w2) w3)] *)

```

LISTING 17. Example code

```

1 g[v2_, v3_, w2_, w3_] := %
2 {equ5, equ6, equ7} // Simplify
3 (* {0, 0, 0} *)

```

Finally, designating the solution for g as *ansg*,

LISTING 18. Example code

```

1 (ansg /. {v2 -> x2/x1, v3 -> x3/x1, w2 -> y2/y1, w3 -> y3/y1}) // Simplify
2 (* C[1][((x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)))/(x1 (x2 y1 + x3 (y1 + y2)) y3
3 )] *)
4 f[x1_, x2_, x3_, y1_, y2_, y3_] := %
5 {equ1, equ2, equ3, equ4}
6 (* {0, 0, 0, 0} *)

```

3.2. Lattice W_4 algebra. .

LISTING 19. Example code

```

1 p = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x1];
2
3 q = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x2];
4
5 r = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x3];
6
7 o = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y1];
8
9 x = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y2];

```

```

10
11 a = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y3];
12
13 b = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z1];
14
15 c = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z2];
16
17 d = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z3];
18
19 equ1 = 2 x1 p + 2 x2 q + 2 x3 r - y1 o - y2 x - y3 a;
20
21 equ2 = -x1 p - x2 q - x3 r - z1 b - z2 c - z3 d + 2 y1 o + 2 y2 x + 2 y3 a;
22
23 equ3 = 2 z1 b + 2 z2 c + 2 z3 d - y1 o - y2 x - y3 a;
24
25 equ4 = (x1 (x1 + 2 x2 + 2 x3)) p + (x2 (x2 +
26 2 x3)) q + (x3^2) r - (y1 (x2 + x3)) o - y2 x3 x;
27
28 equ5 = (y1 (y1 + 2 y2 + 2 y3)) o + (y2 (y2 + 2 y3)) x + (y3^2) a - (x1 (y1 + y2 + y3)) p
29 - x2 (y2 + y3) q - x3 y3 r - z1 (y2 + y3) b - z2 y3 c;
30
31 equ6 = (z1 (z1 + 2 z2 + 2 z3)) b + (z2 (z2 + 2 z3)) c + (z3^2) d - (y1 (z1 + z2 + z3)) o
32 - y2 (z2 + z3) x - y3 z3 a;

```

LISTING 20. Example code

```

1 DSolve[{HX == 0, HY == 0, , HY == 0, EX == 0, EY == 0, EZ == 0},
2 f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
3 z2, z3}]

```

Again *DSolve* returns un-evaluated, meaning that it can not solve the system of equations.

LISTING 21. Example code

```

1 DSolve[ HX + 3 HZ + 2 HY == 0,
2 f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
3 z2, z3}][[1, 1]]
4 (* f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
5 C[1][x1, x2, x3, y1, y2, y3, z2/z1, z3/z1]*)

```

LISTING 22. Example code

```

1 DSolve[ HX + HZ + 2 HY == 0,
2 f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
3 z2, z3}][[1, 1]]
4 (* f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
5 C[1][x1, x2, x3, y2/y1, y3/y1, z1, z2, z3] *)

```

LISTING 23. Example code

```

1 DSolve[ 3 HX + HZ + 2 HY == 0,
2 f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
3 z2, z3}][[1, 1]]
4 (* f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
5 C[1][x2/x1, x3/x1, y1, y2, y3, z1, z2, z3] *)

```

As before, this computation can be simplified by the substitution,

LISTING 24. Example code

```
1 f[x1, x2, x3, y1, y2, y3, z1, z2, z3] := g[x2/x1, x3/x1, y2/y1, y3/y1, z2/z1, z3/z1]
```

in which case the six equations become

LISTING 25. Example code

```
1 Simplify[{equ1, equ2, equ3}]
2 (* {0, 0, 0} *)
3
4 equ4 = Simplify[Simplify[equ4]/x1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
5 y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
6 (* (v2 + v3)*w3*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
7 v2*w2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
8 v3*(1 + 2*v2 + v3)*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
9 v2*(1 + v2)*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
10
11 equ5 = Simplify[Simplify[equ5]/y1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
12 y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
13 (* k3*(w2 + w3)*Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] +
14 k2*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -
15 w3*(1 + 2*w2 + w3)*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -
16 w2*(1 + w2)*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
17 v3*(1 + w2)*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
18 v2*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
19
20 equ6 = Simplify[Simplify[equ6]/z1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
21 y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
22 (* -(k3*(1 + 2*k2 + k3)*Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]) -
23 k2*(1 + k2)*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] +
24 (1 + k2)*w3*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
25 w2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
```

As before, this system of first-order *PDEs* can be solved by using the procedure described in Chapter V, Sec IV of Goursat's Differential Equations. The first step is to find the complete, non-commutative group of differential operators that includes *equ4*, *equ5*, and *equ6*. To do so, we use the function *comm*, generalized from *W₃*

LISTING 26. Example code

```
1 drv = {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
2 Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
3 Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
4 Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3],
5 Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3],
6 Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]};
7 comm[equa_, equb_] := Collect[
8 (equa /. {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, v2],
9 Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, v3],
10 Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, w2],
11 Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, w3],
12 Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, k2],
13 Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] -> D[equb, k3]}) -
14 (equb /. {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, v2],
15 Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, v3],
16 Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, w2],
```

```

17 Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, w3],
18 Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, k2],
19 Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] -> D[equa, k3]],
20 drv, Simplify]
21
22 equ7 = comm[equ4, equ5]
23 (* (k3*v2*w2 + k3*(v2 + v3)*w3)*Derivative[0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]
    +
24 k2*v2*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -
25 w3*(v3*(1 + w2 + w3) + v2*(1 + 2*w2 + w3))*
26 Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -
27 v2*w2*(1 + w2)*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
28 v3*(v3*(1 + w2) + v2*(2 + w2))*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
29 v2^2*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
30
31 equ8 = comm[equ5, equ6]
32 (* -(k3*((1 + 2*k2 + k3)*w2 + (1 + k2 + k3)*w3)*
33 Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]) -
34 k2*(1 + k2)*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] +
35 w3*((2 + k2)*w2 + (1 + k2)*w3)*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
36 w2^2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
37 v3*w2*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)

```

which are independent of the first three operators, increasing the size of the group to five. $\text{comm}[\text{equ4}, \text{equ6}]$ vanishes identically and so does not add an operator. On the other hand, the seven additional commutators involving equ7 and equ8 yield expressions that are linear combinations of $\{\text{equ4}, \text{equ5}, \text{equ6}, \text{equ7}, \text{equ8}\}$. Thus, these five operators comprise the entire group.

From this information alone, we know that g is an arbitrary function of precisely one first integral. This first integral can be obtained by systematically eliminating variables and equations, one pair at a time, until a single equation of two variable remains. Start by solving any one of the equations.

LISTING 27. Example code

```

1 DSolve[equ4 == 0,
2 g[v2, v3, w2, w3, k2, k3], {v2, v3, w2, w3, k2, k3}][[1,
3 1]] // FullSimplify
4 (* g[v2, v3, w2, w3, k2, k3] ->
5 C[1][k2, k3]/((v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2) *)

```

LISTING 28. Example code

```

1 g[v2_, v3_, w2_, w3_, k2_, k3_] :=
2 h[w2, (v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2, k2, k3];
3 tr1 = {equ4, equ5, equ6, equ7, equ8} // Simplify;

```

LISTING 29. Example code

```

1 solw2 = equ4/(v2 w2) // FullSimplify;

```

LISTING 30. Example code

```

1 newvar = Solve[
2 Thread[{b1, b2, b3, b4, b5, b6} == List @@ solw2], {v2, v3, w2,
3 w3, k2, k3}] // Flatten;
4 tr1p = Collect[FullSimplify[Rest[tr1] /. solw2 -> 0 /. newvar], b1,

```

```

5 FullSimplify] /. b1*(z_) -> 0 /.
6 Derivative[0, n1_, n2_, n3_, n4_, n5_][h][b1, b2, b3, b4, b5,
7 b6] -> Derivative[n1, n2, n3, n4, n5][h][b2, b3, b4, b5, b6];

```

LISTING 31. Example code

```

1 DSolve[First@tr1p == 0,
2 h[b2, b3, b4, b5, b6], {b2, b3, b4, b5, b6}] /. Log[z_] -> z;
3 h[b2_, b3_, b4_, b5_, b6_] :=
4 j[b4, b3, b5, (1 + b2 + b3) b4 b6, b6 (1 - b4)];
5 tr2 = tr1p // Simplify;
6 solb4 = First@tr2/(b4 (b4 - 1)) // Simplify;
7 newvar = Solve[
8 Thread[{c1, c2, c3, c4, c5} == List @@ solb4], {b2, b3, b4, b5,
9 b6}] // Flatten;
10
11 tr2p = Collect[(Cancel[(c1 - 1) Rest@tr2] /. solb4 -> 0 /. newvar) //
12 FullSimplify, c1, FullSimplify];
13 tr2p[[3]] = tr2p[[3]]/c1;
14 tr2p = tr2p /. c1 z_ -> 0 /.
15 Derivative[0, n1_, n2_, n3_, n4_][j][c1, c2, c3, c4, c5] ->
16 Derivative[n1, n2, n3, n4][j][c2, c3, c4, c5];

```

LISTING 32. Example code

```

1 DSolve[Last@tr2p == 0, j[c2, c3, c4, c5], {c2, c3, c4, c5}];
2 j[c2_, c3_, c4_, c5_] := l[c5, c2, c3, (c2 - c4)/(1 + c3 + c5)];
3 tr3 = -tr2p // Simplify // RotateRight;
4 solc5 = First@tr3/(c5 (1 + c3 + c5)) // Simplify;
5 newvar = Solve[
6 Thread[{d1, d2, d3, d4} == List @@ solc5], {c2, c3, c4, c5}] //
7 Flatten;
8 tr3p = Collect[(Rest@tr3 /. solc5 -> 0 /. newvar) // FullSimplify, d1,
9 FullSimplify] /. d1 z_ -> 0 /.
10 Derivative[0, n1_, n2_, n3_][l][d1, d2, d3, d4] ->
11 Derivative[n1, n2, n3][l][d2, d3, d4];

```

LISTING 33. Example code

```

1 DSolve[Last@tr3p == 0, l[d2, d3, d4], {d2, d3, d4}] // Simplify;
2 l[d2_, d3_, d4_] := m[d3, (1 + d2) d3, (d2 (1 + d2 - d4))/d4];
3 tr4 = tr3p // Simplify // RotateRight;
4 sold3 = First@tr4/(d2 d3);
5 newvar = Solve[Thread[{e1, e2, e3} == List @@ sold3], {d2, d3, d4}] //
6 Flatten;
7 tr4p = Collect[(-e2/e1) Rest@tr4 /. sold3 -> 0 /. newvar] //
8 FullSimplify, e1, FullSimplify] /. e1 z_ -> 0 /.
9 Derivative[0, n1_, n2_][m][e1, e2, e3] ->
10 Derivative[n1, n2][m][e2, e3];

```

LISTING 34. Example code

```

1 (DSolve[Last@tr4p == 0, m[e2, e3], {e2, e3}] // Simplify) /.
2 Log[z_] -> z;
3 (((((((((%[[1, 1, 2]] /. Thread[{e1, e2, e3} -> List @@ sold3]) //
4 Simplify) /.
5 Thread[{d1, d2, d3, d4} -> List @@ solc5]) //
6 Simplify) /.
7 Thread[{c1, c2, c3, c4, c5} -> List @@ solb4]) //

```

```

8 Simplify) /.
9 Thread[{b1, b2, b3, b4, b5, b6} -> List @@ solw2] //
10 Simplify) /. {v2 -> x2/x1, v3 -> x3/x1, w2 -> y2/y1, w3 -> y3/y1,
11 k2 -> z2/z1, k3 -> z3/z1} // Simplify

```

Final solution

LISTING 35. Example code

```

1 C[1][-(x2 y2 z2 + x1 (y2 z2 + y1 (z1 + z2))) (x3 y3 z3 +
2 x2 (y3 z3 + y2 (z2 + z3)))]/(
3 x2 y2 z2 (x3 y3 z3 + x2 (y3 z3 + y2 (z2 + z3)) +
4 x1 (y3 z3 + y2 (z2 + z3) + y1 (z1 + z2 + z3))))]

```

3.3. Expressing a fractional multivariate polynomials to its low-order polynomial decomposition. Suppose we have given the following question.

Question:

Let f_2 be fractional multivariate polynomial as follows

LISTING 36. Example code

```

1 f2 = -((2 x1 x2 x3 x4 y1 y2^2 y3 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3) (x3 y3 + x2 (y2 +
   y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2))^2 (x2 y1 + x3 (y1 + y2)) (x3 y3
   + x2 (y2 + y3)) (x3 y2 + x4 (y2 + y3))^2));

```

and also let k_1 and k_2 be given as follows

LISTING 37. Example code

```

1 k1 = (x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2)) (x3 y3 +
   x2 (y2 + y3)));
2 k2 = (x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3))/((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (
   y2 + y3)));

```

then express f_2 as a low - order polynomial in k_1 and k_2 .

This can be done as follows.

First, generate a generic low order polynomial.

LISTING 38. Example code

```

1 Map[t1^First@# t2^Last@# &, Tuples[Range[0, 3], 2]].Table[Unique["c"], {16}]
2 (* c3 + c7 t1 + c11 t1^2 + c15 t1^3 + c4 t2 + c8 t1 t2 + c12 t1^2 t2 +
3 c16 t1^3 t2 + c5 t2^2 + c9 t1 t2^2 + c13 t1^2 t2^2 + c17 t1^3 t2^2 +
4 c6 t2^3 + c10 t1 t2^3 + c14 t1^2 t2^3 + c18 t1^3 t2^3 *)

```

and then use SolveAlways. After about twenty seconds we will get result

LISTING 39. Example code

```

1 Flatten@SolveAlways[f2 == (% /. {t1 -> k1, t2 -> k2}), {x1, x2, x3, x4, y1, y2, y3}]
2 (* {c3 -> 0, c4 -> 0, c5 -> 0, c6 -> 0, c11 -> 0, c15 -> 0, c7 -> 0, c12 -> 2, c16
   -> 0, c8 -> -2, c10 -> 0, c13 -> -2, c14 -> 0, c17 -> 0, c18 -> 0, c9 ->
   2} *)

```

And we have the final solution

LISTING 40. Example code

```

1 Factor[% /. %]
2 (* -2 (-1 + t1) t1 (-1 + t2) t2 *)

```

which is the desired result.
And for completeness we have

LISTING 41. Example code

```
1 Simplify[f2 == % /. {t1 -> k1, t2 -> k2}]
2 (* True *)
```

Also here we have much faster alternative:

Because SolveAlways determines the coefficients c for any $\{x_1, x_2, x_3, x_4, y_1, y_2, y_3\}$, Solve must be able to obtain the same values for the coefficients c for specific values of $\{x_1, x_2, x_3, x_4, y_1, y_2, y_3\}$, and much faster. As before we do have f_2 and k_1 and k_2 .

LISTING 42. Example code

```
1 f2 = -((2 x1 x2 x3 x4 y1 y2^2 y3 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3) (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3))) / ((x2 y2 + x1 (y1 + y2))^2 (x2 y1 + x3 (y1 + y2)) (x3 y3 + x2 (y2 + y3)) (x3 y2 + x4 (y2 + y3))^2));
```

LISTING 43. Example code

```
1 k1 = (x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3))) / ((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3)));
2 k2 = (x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3)) / ((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (y2 + y3)));
```

LISTING 44. Example code

```
1 tp = Tuples[Range[0, 3], 2]; tp // Length
2 (* 16 *)
```

LISTING 45. Example code

```
1 gp = Map[t1^#[[1]] t2^#[[2]] &, tp].Table[Unique["c"], {tp // Length}]
2 (* c3 + c7 t1 + c11 t1^2 + c15 t1^3 + c4 t2 + c8 t1 t2 + c12 t1^2 t2 + c16 t1^3 t2 + c5 t2^2 + c9 t1 t2^2 + c13 t1^2 t2^2 + c17 t1^3 t2^2 + c6 t2^3 + c10 t1 t2^3 + c14 t1^2 t2^3 + c18 t1^3 t2^3 *)
```

LISTING 46. Example code

```
1 Flatten@Solve[Table[(f2 == (gp /. {t1 -> k1, t2 -> k2})) /. Thread[{x1, x2, x3, x4, y1, y2, y3} -> RandomInteger[{1, 7}, 7]], {n, tp // Length}], List @@ (First@# & /@ (gp /. gp[[1]] -> gp[[1]] z))]
2 (* {c10 -> 0, c11 -> 0, c12 -> 2, c13 -> -2, c14 -> 0, c15 -> 0, c16 -> 0, c17 -> 0, c18 -> 0, c3 -> 0, c4 -> 0, c5 -> 0, c6 -> 0, c7 -> 0, c8 -> -2, c9 -> 2} *)
```

LISTING 47. Example code

```
1 Factor[%% /. %]
2 (* -2 (-1 + t1) t1 (-1 + t2) t2 *)
```

LISTING 48. Example code

```
1 Simplify[f2 == % /. {t1 -> k1, t2 -> k2}]
2 (* True *)
```

Question:

Let f_6 be fractional multivariate polynomial as follows

LISTING 49. Example code

```
1 f6 = (2 x1 x2 x5 x6 y2 (x2 y1 + x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5))) / ((x2 y2 +
  x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4 +
  y5)));
```

and also let k_1, k_2, k_3, k_4, k_5 and k_6 be given as follows

LISTING 50. Example code

```
1 k1 = ((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))) / (x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (
  y1 + y2 + y3)));
2 k2 = ((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (y2 + y3))) / (x3 y2 (x2 y1 + (x3 + x4) (y1 + y2
  ) + x4 y3));
3 k3 = ((x3 y3 + x2 (y2 + y3)) (x4 y4 + x3 (y3 + y4))) / (x3 y3 (x4 y4 + x3 (y3 + y4) + x2
  (y2 + y3 + y4)));
4 k4 = ((x3 y2 + x4 (y2 + y3)) (x4 y3 + x5 (y3 + y4))) / (x4 y3 (x3 y2 + (x4 + x5) (y2 + y3)
  + x5 y4));
5 k5 = ((x4 y4 + x3 (y3 + y4)) (x5 y5 + x4 (y4 + y5))) / (x4 y4 (x5 y5 + x4 (y4 + y5) + x3 (
  y3 + y4 + y5)));
6 k6 = ((x4 y3 + x5 (y3 + y4)) (x5 y4 + x6 (y4 + y5))) / (x5 y4 (x4 y3 + (x5 + x6) (y3 + y4)
  + x6 y5));
```

then express f_6 as a low - order polynomial in k_1, k_2, k_3, k_4, k_5 and k_6 .

LISTING 51. Example code

```
1 tp = Tuples[Range[-1, 1], 6]; tp // Length
2 (* 729 *)
```

LISTING 52. Example code

```
1 gp = Map[t1^#[[1]] t2^#[[2]] t3^#[[3]] t4^#[[4]] t5^#[[5]] t6^#[[6]] &, tp].Table[
  Unique["c"], {tp // Length}];
```

LISTING 53. Example code

```
1 sol = Flatten@ Solve[Table[f6 == (gp /. {t1 -> k1, t2 -> k2, t3 -> k3, t4 -> k4, t5
  -> k5, t6 -> k6})] /. Thread[{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5} ->
  RandomInteger[{1, 11}, 11]], {n, tp // Length}], List @@ (First@# & /@ (gp /. gp
  [[1]] -> gp[[1]] z)); sol /. Rule[_, 0] -> Nothing
2 (* {Nothing, ..., Nothing, c114 -> -2, c115 -> 2, Nothing, ..., Nothing, c123 -> 2,
  c124 -> -2, Nothing, ..., Nothing, c330 -> -2, c331 -> 2, Nothing, ..., Nothing,
  c339 -> 2, Nothing, c340 -> -2, Nothing, ..., Nothing, c357 -> 2, c358 -> -2,
  Nothing, ..., Nothing, c366 -> -2, c367 -> 2, Nothing, ..., Nothing, c87 -> 2, c88
  -> -2, Nothing, ..., Nothing, c96 -> -2, c97 -> 2, Nothing, Nothing} *)
```

LISTING 54. Example code

```
1 Factor[gp /. sol]
2 (* (2 (-1 + t1) (-1 + t3) (-1 + t4) (-1 + t6)) / (t1 t3 t4 t6) *)
```

Which is the desired result.

And as before for completeness we have

LISTING 55. Example code

```
1 Simplify[f6 == % /. {t1 -> k1, t2 -> k2, t3 -> k3, t4 -> k4, t5 -> k5, t6 -> k6}]
2 (* True *)
```

By using Groebner Basis:

Also there is another way for to reach to the solution by using Groebner Basis. But this approach is very slow!

LISTING 56. Example code

```
1 poly = (2 x1 x2 x5 x6 y2 (x2 y1 + x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5))) / ((x2 y2
+ x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4
+ y5)));
```

LISTING 57. Example code

```
1 eqns = {K1 == ((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))) / (x2 y2 (x3 y3 + x2 (y2 +
y3) + x1 (y1 + y2 + y3))),
2 K2 == ((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (y2 + y3))) / (x3 y2 (x2 y1 + (x3 + x4) (y1 +
y2) + x4 y3)),
3 K3 == ((x3 y3 + x2 (y2 + y3)) (x4 y4 + x3 (y3 + y4))) / (x3 y3 (x4 y4 + x3 (y3 + y4) + x2
(y2 + y3 + y4))),
4 K4 == ((x3 y2 + x4 (y2 + y3)) (x4 y3 + x5 (y3 + y4))) / (x4 y3 (x3 y2 + (x4 + x5) (y2 +
y3) + x5 y4)),
5 K5 == ((x4 y4 + x3 (y3 + y4)) (x5 y5 + x4 (y4 + y5))) / (x4 y4 (x5 y5 + x4 (y4 + y5) +
x3 (y3 + y4 + y5))),
6 K6 == ((x4 y3 + x5 (y3 + y4)) (x5 y4 + x6 (y4 + y5))) / (x5 y4 (x4 y3 + (x5 + x6) (y3 +
y4) + x6 y5)));
```

Now let us compute Groebner Basis

LISTING 58. Example code

```
1 gb = GroebnerBasis[eqns, {x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6}];
```

The remainder r gives a representation of poly in terms of $K1$, $K2$, $K3$, $K4$, $K5$ and $K6$.

LISTING 59. Example code

```
1 {qs, r} = PolynomialReduce[poly, gb, {x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6}];
```

Where r is our solution in $K1$, $K2$, $K3$, $K4$, $K5$ and $K6$. And the following code validates correctness:

LISTING 60. Example code

```
1 poly == r /. ToRules[And @@ eqns] // Expand
```

And please note that, this may take a while. (May be more than a while! It depends on how powerful is your computer.)

3.4. Checking symmetries in our shift operators. . First, before starting, we need to know which variables are employed in our functions. For to do this we employ the following code:

Set

LISTING 61. Example code

```
1 f9 = (2 x1 x2 x5 y2 y5 y6 z2 (x2 y1 y2 z1 + x2 y1 y3 z1 + x3 y1 y3 z1 + x2 y1 y3 z2 + x3
y1 y3 z2 + x3 y2 y3 z2) z3^2 z4 (x4 x5 y4 z4 + x4 x6 y4 z4 + x4 x6 y5 z4 + x4 x6
y4 z5 + x4 x6 y5 z5 + x5 x6 y5 z5)) / ((x1 y1 z1 + x1 y1 z2 + x1 y2 z2 + x2 y2 z2) (
x2 y2 z2 + x2 y2 z3 + x2 y3 z3 + x3 y3 z3)^2 (x4 y4 z3 + x4 y5 z3 + x5 y5 z3 + x5
y5 z4)^2 (x5 y5 z4 + x5 y6 z4 + x6 y6 z4 + x6 y6 z5));
```

and

LISTING 62. Example code

```

1 k1 = ((x1 y1 z1 + x2 y2 z2 + x1 (y1 + y2) z2) (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3))
      /((x2 y2 z2 (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3 + x1 (y2 z2 + (y2 + y3) z3 + y1 (
      z1 + z2 + z3)))));
2 k2 = (((x2 + x3) y1 z1 + x3 (y1 + y2) z2) ((x3 + x4) y2 z2 + x4 (y2 + y3) z3))/((x3 y2 z2
      (x2 y1 z1 + (x3 + x4) (y2 z2 + y1 (z1 + z2)) + x4 (y1 + y2 + y3) z3));
3 k3 = ((x2 (y2 + y3) z1 + x3 y3 (z1 + z2)) (x3 (y3 + y4) z2 + x4 y4 (z2 + z3)))/(x3 y3 z2
      (x2 (y2 + y3 + y4) z1 + x3 (y3 + y4) (z1 + z2) + x4 y4 (z1 + z2 + z3)));
4 k4 = ((x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3) (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4))
      /((x3 y3 z3 (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4 + x2 (y3 z3 + (y3 + y4) z4 + y2
      (z2 + z3 + z4)))));
5 k5 = (((x3 + x4) y2 z2 + x4 (y2 + y3) z3) ((x4 + x5) y3 z3 + x5 (y3 + y4) z4))/((x4 y3 z3
      (x3 y2 z2 + (x4 + x5) (y3 z3 + y2 (z2 + z3)) + x5 (y2 + y3 + y4) z4));
6 k6 = ((x3 (y3 + y4) z2 + x4 y4 (z2 + z3)) (x4 (y4 + y5) z3 + x5 y5 (z3 + z4)))/(x4 y4
      z3 (x3 (y3 + y4 + y5) z2 + x4 (y4 + y5) (z2 + z3) + x5 y5 (z2 + z3 + z4)));
7 k7 = ((x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4) (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5))/
      (x4 y4 z4 (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5 + x3 (y4 z4 + (y4 + y5) z5 + y3 (z3
      + z4 + z5))));
8 k8 = (((x4 + x5) y3 z3 + x5 (y3 + y4) z4) ((x5 + x6) y4 z4 + x6 (y4 + y5) z5))/((x5 y4 z4
      (x4 y3 z3 + (x5 + x6) (y4 z4 + y3 (z3 + z4)) + x6 (y3 + y4 + y5) z5));
9 k9 = ((x4 (y4 + y5) z3 + x5 y5 (z3 + z4)) (x5 (y5 + y6) z4 + x6 y6 (z4 + z5)))/(x5 y5 z4
      (x4 (y4 + y5 + y6) z3 + x5 (y5 + y6) (z3 + z4) + x6 y6 (z3 + z4 + z5)));

```

Then by using the following code we will obtain the set of our variables which have been employed

LISTING 63. Example code

```

1 Union[Cases[#, _Symbol, Infinity]] & /@ {f9}
2 (* {{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, z1, z2, z3, z4, z5}} *)

```

LISTING 64. Example code

```

1 Union[Cases[#, _Symbol, Infinity]] & /@ {k1, k2, k3, k4, k5, k6, k7, k8, k9}
2 (* {{x1, x2, x3, y1, y2, y3, z1, z2, z3}, {x2, x3, x4, y1, y2, y3, z1, z2, z3}, {x2, x3,
      x4, y2, y3, y4, z1, z2, z3}, {x2, x3, x4, y2, y3, y4, z2, z3, z4}, {x3, x4, x5, y2
      , y3, y4, z2, z3, z4}, {x3, x4, x5, y3, y4, y5, z2, z3, z4}, {x3, x4, x5, y3, y4,
      y5, z3, z4, z5}, {x4, x5, x6, y3, y4, y5, z3, z4, z5}, {x4, x5, x6, y4, y5, y6, z3,
      z4, z5}} *)

```

Now in what comes below, we specifically mean that for example in sl_4 in a process for finding $F_9^{(4)}$, the substitution

$$\{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, z1, z2, z3, z4, z5\}$$

instead of

$$\{y6, y5, y4, y3, y2, y1, x6, x5, x4, x3, x2, x1, z5, z4, z3, z2, z1\}$$

transforms $\tau_9^{(4)}$ to $\tau_1^{(4)}$, $\tau_8^{(4)}$ to $\tau_2^{(4)}$, $\tau_7^{(4)}$ to $\tau_3^{(4)}$, and $\tau_6^{(4)}$ to $\tau_4^{(4)}$ while leaving $F_9^{(4)}$ unchanged.

Therefore, those four pairs must enter the expression for $F_9^{(4)}$ symmetrically.

As a result, the generic polynomials we have been using above, can be reduced greatly in numbers of terms, a factor of $(\frac{2}{3})^4$. Corresponding running time then should be reduced by a factor of $(\frac{2}{3})^8$, other things being equal.

It is possible that additional symmetries exist! It needs to be checked!

Here for simplification and for to be able in coding them, we write instead $F_9^{(4)} := F9$ and $\tau_7^{(4)} := Ki$;

LISTING 65. Example code

```
1 arg1 = {a1, a2, a3, a4, a5, a6, b1, b2, b3, b4, b5, b6, d1, d2, d3, d4, d5};
2 arg2 = {a6, a5, a4, a3, a2, a1, b6, b5, b4, b3, b2, b1, d5, d4, d3, d2, d1};
3 arg3 = {b6, b5, b4, b3, b2, b1, a6, a5, a4, a3, a2, a1, d5, d4, d3, d2, d1};
```

LISTING 66. Example code

```
1 F9[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := (2 x1 x2 x5 y2 y5 y6 z2 (x2 y1 y2 z1 + x2 y1 y3 z1 +
2 x3 y1 y3 z2 + x2 y1 y3 z2 + x3 y2 y3 z2) z3^2 z4 (x4 x5 y4 z4 + x4 x6 y4
   z4 + x4 x6 y5 z4 + x4 x6 y4 z5 + x4 x6 y5 z5 + x5 x6 y5 z5)) / ((x1 y1 z1 + x1 y1 z2
   + x1 y2 z2 + x2 y2 z2) (x2 y2 z2 + x2 y2 z3 + x2 y3 z3 + x3 y3 z3)^2 (x4 y4 z3 + x4
   y5 z3 + x5 y5 z3 + x5 y5 z4)^2 (x5 y5 z4 + x5 y6 z4 + x6 y6 z4 + x6 y6 z5))
```

LISTING 67. Example code

```
1 Simplify[F9 @@ arg1 == F9 @@ arg3]
2 (* True *)
```

LISTING 68. Example code

```
1 K1[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := (((x1 y1 z1 + x2 y2 z2 + x1 (y1 + y2) z2) (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3)
   )) / (x2 y2 z2 (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3 + x1 (y2 z2 + (y2 + y3) z3 + y1
   (z1 + z2 + z3))));
2 K2[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := (((x2 + x3) y1 z1 + x3 (y1 + y2) z2) ((x3 + x4) y2 z2 + x4 (y2 + y3) z3)) / (x3 y2
   z2 (x2 y1 z1 + (x3 + x4) (y2 z2 + y1 (z1 + z2)) + x4 (y1 + y2 + y3) z3));
3 K3[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := ((x2 (y2 + y3) z1 + x3 y3 (z1 + z2)) (x3 (y3 + y4) z2 + x4 y4 (z2 + z3))) / (x3 y3
   z2 (x2 (y2 + y3 + y4) z1 + x3 (y3 + y4) (z1 + z2) + x4 y4 (z1 + z2 + z3)));
4 K4[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := ((x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3) (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4)
   z4)) / (x3 y3 z3 (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4 + x2 (y3 z3 + (y3 + y4) z4 +
   y2 (z2 + z3 + z4))));
5 K5[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := (((x3 + x4) y2 z2 + x4 (y2 + y3) z3) ((x4 + x5) y3 z3 + x5 (y3 + y4) z4)) / (x4 y3
   z3 (x3 y2 z2 + (x4 + x5) (y3 z3 + y2 (z2 + z3)) + x5 (y2 + y3 + y4) z4));
6 K6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := ((x3 (y3 + y4) z2 + x4 y4 (z2 + z3)) (x4 (y4 + y5) z3 + x5 y5 (z3 + z4))) / (x4
   y4 z3 (x3 (y3 + y4 + y5) z2 + x4 (y4 + y5) (z2 + z3) + x5 y5 (z2 + z3 + z4)));
7 K7[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := ((x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4) (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5)
   ) / (x4 y4 z4 (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5 + x3 (y4 z4 + (y4 + y5) z5 + y3 (
   z3 + z4 + z5))));
8 K8[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := (((x4 + x5) y3 z3 + x5 (y3 + y4) z4) ((x5 + x6) y4 z4 + x6 (y4 + y5) z5)) / (x5 y4
   z4 (x4 y3 z3 + (x5 + x6) (y4 z4 + y3 (z3 + z4)) + x6 (y3 + y4 + y5) z5));
9 K9[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
   := ((x4 (y4 + y5) z3 + x5 y5 (z3 + z4)) (x5 (y5 + y6) z4 + x6 y6 (z4 + z5))) / (x5 y5
   z4 (x4 (y4 + y5 + y6) z3 + x5 (y5 + y6) (z3 + z4) + x6 y6 (z3 + z4 + z5)));
```

LISTING 69. Example code

```

1 Simplify[K1 @@ arg1 == K9 @@ arg3]
2 (* True *)

```

LISTING 70. Example code

```

1 Simplify[K3 @@ arg1 == K7 @@ arg3]
2 (* True *)

```

LISTING 71. Example code

```

1 Simplify[K4 @@ arg1 == K6 @@ arg3]
2 (* True *)

```

Checking symmetries in F6:

We can find the set of variables in a same way as what we did for $F_9^{(4)}$ and so here we omit most of the calculations.

Again as in $F_9^{(4)}$, here we specifically mean that the substitution $\{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5\}$ instead of $\{x6, x5, x4, x3, x2, x1, y5, y4, y3, y2, y1\}$ transforms $\tau_6^{(4)}$ to $\tau_1^{(4)}$, $\tau_5^{(4)}$ to $\tau_2^{(4)}$, $\tau_4^{(4)}$ to $\tau_3^{(4)}$ while leaving $F_6^{(4)}$ unchanged.

Therefore, those three pairs must enter the expression for $F_6^{(4)}$ symmetrically.

LISTING 72. Example code

```

1 arg1 = {a1, a2, a3, a4, a5, a6, b1, b2, b3, b4, b5};
2 arg2 = {a6, a5, a4, a3, a2, a1, b5, b4, b3, b2, b1};

```

LISTING 73. Example code

```

1 F6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := (2 x1 x2 x5 x6 y2 (x2 y1 +
  x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5)))/((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (
  y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4 + y5)));

```

LISTING 74. Example code

```

1 F6 @@ arg1 == F6 @@ arg2
2 (* True *)

```

LISTING 75. Example code

```

1 K1[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x2 y2 + x1 (y1 + y2)) (x3
  y3 + x2 (y2 + y3)))/(x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)));
2 K2[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x2 y1 + x3 (y1 + y2)) (x3
  y2 + x4 (y2 + y3)))/(x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3));
3 K3[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x3 y3 + x2 (y2 + y3)) (x4
  y4 + x3 (y3 + y4)))/(x3 y3 (x4 y4 + x3 (y3 + y4) + x2 (y2 + y3 + y4)));
4 K4[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x3 y2 + x4 (y2 + y3)) (x4
  y3 + x5 (y3 + y4)))/(x4 y3 (x3 y2 + (x4 + x5) (y2 + y3) + x5 y4));
5 K5[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x4 y4 + x3 (y3 + y4)) (x5
  y5 + x4 (y4 + y5)))/(x4 y4 (x5 y5 + x4 (y4 + y5) + x3 (y3 + y4 + y5)));
6 K6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x4 y3 + x5 (y3 + y4)) (x5
  y4 + x6 (y4 + y5)))/(x5 y4 (x4 y3 + (x5 + x6) (y3 + y4) + x6 y5));

```

LISTING 76. Example code

```
1 Simplify[K1 @@ arg1 == K6 @@ arg2]
2 (* True *)
```

LISTING 77. Example code

```
1 Simplify[K2 @@ arg1 == K5 @@ arg2]
2 (* True *)
```

LISTING 78. Example code

```
1 Simplify[K3 @@ arg1 == K4 @@ arg2]
2 (* True *)
```

```
5 1
```

```
6 2
```

⁵Farrokh Razavinia, Department of discrete mathematics, Moscow Institute of Physics and Technology, Institutskiy per., 9, Dolgoprudny, Moscow Oblast, Russia

⁶Brendan B. Godfrey, Institute for Research in Electronics and Applied Physics (The University of Maryland), College Park, United States of America